Interaction Patterns in Computer-assisted Semantic Annotation of Text
An Empirical Evaluation

Jaroslav Dytrych and Pavel Smrz
Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence, Bozetechova 2, 612 66 Brno, Czech Republic

Keywords: Computer-assisted Tagging, Semantic Annotation, Event Extraction.

Abstract: This paper examines user interface options and interaction patterns evinced in tools for computer-assisted semantic enrichment of text. It focuses on advanced annotation tasks such as hierarchical annotation of complex relations and linking entities with highly ambiguous names and explores how decisions on particular aspects of annotation interfaces influence the speed and the quality of computer-assisted human annotation processes. Reported experiments compare the 4A annotation system, designed and implemented by our team, to RDFaCE and GATE tools that all provide advanced annotation functionality. Results show that users are able to reach better consistency of event annotations in less time when using the 4A editor. A set of experiments is then conducted that employ 4A's high flexibility and customizability to find an optimal amount of displayed information and its presentation form to reach best results in linking entities with highly ambiguous names. The last set of experiments then proves that 4A's particular way of implementing the concept of semantic filtering speeds up event annotation processes and brings higher consistency when compared to alternative approaches.

1 INTRODUCTION

Semantic annotation of text has become a popular topic of the natural language processing in recent years. Various tools have been developed that automatically enrich text with semantic annotations. However, the quality of results achieved using fully automatic approaches varies significantly across annotation tasks and input data. It can be particularly low for complex and highly ambiguous cases Moro and Navigli (2015); Surdeanu and Heng (2014).

There is still a room for computer-assisted manual annotation. Although it would not be wise to rely on user tagging of extremely large amounts of textual data available on the current web, manual annotation is still used in the preparation of training data for supervised machine learning methods, for annotation validation tasks, constructing meaning through annotations, etc. To support the annotation process, various semantic editors, crowdsourcing plugins for existing tools and web browser annotation extensions have been created Ciccarese et al. (2012); Grassi et al. (2013); Handschuh et al. (2002); Heese et al. (2010). Also, general linguistic annotation tools such as BRAT Stenetorp et al. (2012) are frequently used for manual creation of annotated data.

The high number of existing annotation systems contrasts with the fact that there are very few studies comparing particular features of the tools and discussing their suitability for specific tasks (some of them are briefly reviewed in Section 2 – Related work). To the best of our knowledge, no existing study compares design patterns employed in such tools that are relevant for complex annotation of events and disambiguation of highly ambiguous entities. This is the main purpose of the set of experiments conducted by our team.

People understand the term *event* intuitively as *who did what to whom when and where*. However, definitions of the term in papers on event extraction vary according to the focus their authors give to particular aspects of the problem in hand. For example, an event can mean "a change on the state of a molecule or molecules" in biomedical text mining systems Kim et al. (2009). We simply adopt the intuitive natural definition in our work and take events as situations, actions or occurrences that happen or take place at a certain location in a certain time. Having the text mining focus in mind, events can be represented as a complex combination of relations linked to a set of empirical observations from texts Hogenboom et al. (2011).

Initial experiments reported in this paper compare three semantic annotation tools – GATE Cun-

Dytrych, J. and Smrz, P.
Interaction Patterns in Computer-assisted Semantic Annotation of Text - An Empirical Evaluation.
DOI: 10.5220/0005695900740084
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 2*, pages 74-84
ISBN: 978-989-758-172-4

ningham et al. (2011), RDFaCE Khalili et al. (2012), and 4A Smrz and Dytrych (2011, 2015) on the task of hierarchical annotation of complex relations in text. The annotation process consisted of selecting parts of a text corresponding to an event of a specific type, filling its attributes (slots) by entities and values mentioned in the text, and disambiguating the entities by linking them to a reference resource (mostly DBPedia/Wikipedia). Results of the user study with 6 participants indicate that different approaches represented by the three tested tools lead to different quality of the annotation and different times to finish the task.

Other two sets of experiments then employ the 4A system in a study confronting various settings of the annotation user interface and corresponding interaction patterns. It is shown that the commonly used practice of annotation tools asking people to disambiguate entities based just on a suggested type and a displayed URL leads to a poor quality of results. The opposite approach giving users immediately extensive information on entities potentially corresponding to an ambiguous name not only leads to longer annotation times but, surprisingly, does not bring the best annotation accuracy either. A trade-off between the insufficient and the excessive is found and demonstrated to provide the most efficient setting. Experiments also call attention to semantic filtering and other enhancements of advanced user interfaces by showing their impact on annotation consistency and speed.

The rest of the paper is organized as follows: After Related work, Section 3 discusses high variability of text annotation tools and various factors that can influence comparison results. Research questions and experiments run to answer them are presented in Section 4. The last section concludes the study and summarizes its results.

2 RELATED WORK

As mentioned above, studies comparing user experiences with tools for semi-automatic text annotation are rare. The Knowledge Web project benchmarked 6 textual annotation tools considering various criteria including usability (installation, documentation, aesthetics…), accessibility (user interface features), and interoperability (platforms and formats) Maynard et al. (2007). Most of the parameters are out of scope of our study but at least some of them are covered in an open shared document[1] that we prepared to compare usage characteristics of annotation tools from the perspective of complex annotation tasks.

Maynard (2008) compares annotation tools from the perspective of a manual annotator, an annotation consumer, a corpus developer, and a system developer. Although the study is already 7 years old, most evaluation criteria are still valid. The study partially influenced our work.

Yee (2002) motivates the implementation of CritLink – a tool enabling users to attach annotations to any location on any public web page – by a table summarizing shortcomings of existing web tools. As opposed to our approach, the comparison focuses on basic annotation tasks only and stresses technical aspects rather than the user experience.

Similarly to other studies, Reeve and Han (2005) compare semantic annotation platforms focusing on the performance of background pre-annotation components that generate annotation suggestions. As modern user interaction tools can freely change the back-ends and generate suggestions by a range of existing annotation systems, the work is relevant only for a historical perspective.

3 COMPARING SEMANTIC ANNOTATORS – APPLES AND ORANGES?

When planning an experimental evaluation of semantic annotation frameworks, one has to take into account features significantly differing across the tools as well as varying aspects of the annotation process that can influence results. This is particularly true if the comparison involves the annotation time and the quality of created annotations. The following paragraphs point out several risks that could negatively influence reliability of such a comparison and strategies to mitigate the risks followed in the reported experimental work.

Three levels of potential problems can be distinguished:

- specificity of the annotation tasks performed in the experiments;
- incommensurability of the tools themselves;
- varying experience and expertise of testers.

Computer-assisted semantic annotation refers to a wide range of tasks. It can involve just a simple identification of entity mentions of few specific types in a text, but also full linking of potentially ambiguous entity names to a background knowledge base, annotation of complex hierarchical relations and their in-

[1] https://docs.google.com/spreadsheets/d/14ionbRVYBQuD0cNLazKfRWYzrkax3qFCspm9SiaG5Aw

dividual attributes. Domain of the text being annotated (e.g., biomedical v. general), its genre, register, or source (for example, news articles v. tweets) may also vary across annotation experiments – they can require particular approaches to text pre-processing and can imply different results of the automatic pre-annotation.

Datasets to be annotated can correspond to a representative subset of texts or they can focus on a chosen phenomenon and mix data accordingly. This variance can be demonstrated by differing nature of datasets employed in previous annotation challenges. For example, the short text track of the 2014 Entity Recognition and Disambiguation (ERD) Challenge[2] stressed limited contexts that naturally appear in web search queries from past TREC competitions. On the other hand, the SemEval-2015 Task 10[3] particularly deals with annotations relevant for sentiment analysis in microblog (Twitter) messages. The Entity Discovery and Linking (EDL) track at NIST TAC-KBP2015[4] then aims to extract named entity mentions, including person nominal mentions, link them to an existing Knowledge Base (KB) and cluster mentions for entities that do not have corresponding KB entries. Obviously, the degree of ambiguity of entities mentioned in annotated texts as well as proportions of occurrences corresponding to particular meanings can have a crucial impact on the speed and accuracy of the annotation process.

Experiments reported in this paper involve annotation of general web page texts (from the CommonCrawl corpus[5] – see below). Initial ones take random sentences based on trigger words (see Section 4.2 for details). Remaining experiments focus on entity linking tasks that are particularly difficult for automatic tools due to the ambiguity of names. We believe that making people annotate occurrences for which automatic tools often fail makes the scenario of the manual annotation tasks more realistic. Sentences to be annotated are particularly selected to guarantee that there is at least one example of an occurrence corresponding to each potential meaning covered by the knowledge base. To evaluate a realistic setting, a part of the dataset is also formed by mentions not covered by the reference resources used.

Various aspects of annotation interfaces also influence experimental results. Some annotation tools aim at general applicability for semantic processes. Others are particularly tailored for paid-crowd annotation scenarios Bontcheva et al. (2014) so that they can be unsuitable for collaborative environments. Also, tools can be tied up with a particular annotation back-end or they can be only loosely coupled with a (preferred) annotator tool that can be easily changed or extended for specific tasks. Other features of annotation tools, especially those related to user interfaces and interaction patterns, are briefly discussed in the following section.

Skills, a current state of mind and motivation of users participating in experiments can also influence results. Measured quality and the time always need to be interpreted with respect to these aspects. It can be expected that users with an experience in using a particular tool will better understand its user interface and will be able to achieve better results using the tool. Also, expertise in a domain in question can speed up the annotation process, especially the disambiguation of specialized entity mentions.

Experimental settings that can award quality over quantity or vice versa can lead to dramatically different times and amounts of annotation errors. Indeed, users in our experiments realized a trade-off between the time spent on each particular case and resulting quality (e.g., users' certainty that they considered enough context to correctly disambiguate an entity mention). While our users asked for preferences in this situation, this finding can be also expected in the paid-for crowdsourcing settings that need to apply sophisticated quality control mechanisms to prevent annotators' temptation to cheat Wang et al. (2013).

Research questions defined in the following section also relate to the amount of information necessary for a correct (and fast) decision on entity disambiguation problems and the way to present potential choices. The number of displayed suggestions and their attributes as well as particular annotation steps followed influence user's attention paid to the task and, consequently, the accuracy of results.

As detailed in Section 4.3, our experiments comparing features of annotation tools were conducted with users that had no previous experience in using the tools and no particular expertise in the field. They had 20 minutes to familiarise themselves with each particular tool (the order in which they tested the tools was unique for all users as there are 6 possible combinations for three tools). Task instructions stressed that users should be as accurate as possible but that time is limited. There was an informal competition on who will deliver the fastest and the most accurate results but no particular incentives (except for a beer for the winner) were promised or given.

[2]http://web-ngram.research.microsoft.com/erd2014/

[3]http://alt.qcri.org/semeval2015/task10/

[4]http://nlp.cs.rpi.edu/kbp/2015/index.html

[5]http://blog.commoncrawl.org

4 ANNOTATION EXPERIMENTS

4.1 Research Questions

Reported experiments aim at answering the following research questions:

1. How design choices of particular annotation tools impact the quality of results and the annotation time.
2. To what extent the amount of information shown to disambiguators influences the results.
3. Whether users benefit from knowledge of potential alternative annotations and confidence levels of provided suggestions.
4. What quality the concept of semantic filtering brings to the annotation process.

Initial annotation experiments address the first question. They compare different user interfaces and interaction patterns as exemplified by three specific annotation systems. Various features that can influence annotation performance need to be considered. Some tools make no visible distinction between pre-annotations generated by a back-end automatic system and manual annotations entered by users. Other tools explicitly distinguish the system suggestions from the accepted or newly entered annotations. This can have an impact on consistency of the annotation, i. e. the number of cases users did not check or just missed a suggestion.

Underlying annotation patterns for events and other complex relations and their attributes vary across tools too. Advanced tools enable defining sophisticated templates and type constraints that control filling of event slots. Of course, systems differ in their actual application of the general approach and the way they implement it influences annotation performance aspects.

Various values entered by users often need to correspond to an entry in a controlled vocabulary or a list of potential items. An example of such a case is a URL linking an entity mention to a reference knowledge source. Tools support entering such values through autocomplete functions that can present not only the value to be entered, but also additional information that helps users to choose the right value. For example, the 4A client shows not only an URL link, but also full names and disambiguation contexts. The RDFaCE, on the other hand, autocompletes just URLs in this context. As shown in the next section, this also impacts the results.

Other two questions mentioned above are covered by the second set of experiments. It is clear that the amount of information shown to the user and its form can influence the speed and accuracy of the annotation process. If the displayed information is not sufficient for a decision what a correct link for an entity is, the user will have to search for additional information. On the other hand, if a tool lets users read more than necessary, the annotation speed can decrease.

Most of the explored systems show just a URL and let the user explore it if she is not sure that the linked information corresponds to an expected one. This can speed up the annotation process but it can also make it error-prone. The 4A system enables filtering displayed information and fine-tuning the way it is shown. Detailed entity attributes can be folded and shown only if the user asks for them.

Without a system support, users are often unaware of ambiguity of some names. It causes no harm for frequent appearances of dominant senses. However, if a user is not an expert in a field where two or more potential links to an underlying resource can appear, she can easily confirm an incorrectly suggested link for an ambiguous name. The risk can be mitigated if the tools let users know about alternatives. The question is how an optimal setting for this function looks like – whether this should be a default behaviour or the system should notify the user only if an automatically computed confidence is smaller than a threshold or the difference between a suggested option and the second one is closer than a threshold. These aspects are discussed in the second experiment too.

The goal of the third set of experiments is then to answer the last research question dealing with the role of semantic filtering. It is not easy to enter complex annotations such as events. Advanced mechanisms suggesting slot filling can make the process more consistent and fast. The 4A tool supports hierarchical annotation which highlights potential nested annotations if an upper-level type is known. As shown at the end of this section, such an approach leads to speedups of the annotation process.

4.2 Data Preparation

Texts to be annotated in the experiments reported in the following subsections were chosen from general web data contained in the CommonCrawl corpus from December 2014.[6] First experiments comparing annotation frameworks dealt with general annotation of events. Text selection did not address any specific objective (in contrast to next experiments) so that contexts containing mentions of named entities recognized by all three tools and a trigger word (verb) corresponding to travels and visits of people to various

[6] http://blog.commoncrawl.org/2014/12/

places were pre-selected. The data was then manually annotated by two authors of this paper, annotation disagreements were solved by choosing correct ones in clear cases and excluding few cases considered unclear.

The second set of experiments, looking into an optimal amount of displayed information, needed data containing ambiguous names with a proportional representation of two or more alternatives. Inspired by WikiLinks[7], we searched the CommonCrawl data for cases linking a name to two or more distinct Wikipedia URLs. To filter out potential interdependencies among various options and enable focusing on key attributes in the first part of experiments, a majority of the prepared dataset consists of pairs of texts mentioning a name shared by two distinct entities. For example, the following sentences are included in the resulting data:

1. *Charles Thomson was a Patriot leader in Philadelphia during the American Revolution and the secretary of the Continental Congress (1774–1789) throughout its existence.*
2. *Charles Thomson's best known work is a satire of Sir Nicholas Serota, Director of the Tate gallery, and Tracey Emin, with whom he was friends in the 1980s.*

A smaller part (34 sets) of the data correspond to sentences containing names that could refer to 5 or more entities in the English Wikipedia. For 7 such cases, the subset was further manually extended to include a text with the same ambiguous name referring to an entity not covered by the Wikipedia. This data was used in the second part of the experiments described in Section 4.4.

The last set of experiments combine annotation of events with disambiguation of entity mentions. Consequently, paragraphs with sentences mentioning ambiguous names linked to the Wikipedia that contain a trigger verb indicating a particular type of an event were retrieved from the CommonCrawl data and validated by the authors. Similarly to the data for the first experiments, the dataset was formed from cases in which the pre-annotation process had led to a clear consensus between the annotators. Only 12 texts mentioning events were finally used in the study.

4.3 Comparing Tools

The aim of initial experiments was to compare advanced annotation editors in terms of their interaction patterns and user-interface features that can influence user experience and annotation performance. We were interested whether annotation results obtained by using particular tools will differ in the quality measured by their completeness and accuracy of types of entities filling slots of complex relations and their links to underlying resources (mostly DBPedia/Wikipedia). In addition, times to finish each experiment were measured for each user and then averaged per attribute annotated.

Employed tools represent different approaches to complex annotation tasks (see Figure 1 for examples of event annotation views). The 4A system[8], implemented by our team, pays a special attention to hierarchical annotations and potentially overlapping textual fragments. Users benefit from advanced annotation suggestions and an easy mechanism for entering correct attribute values by simply accepting or rejecting provided suggestions.

The RDFaCE editor[9] is similar to 4A in the way it annotates textual fragments and can be also deployed as an plugin for JavaScript WYSIWYG editor TinyMCE. It can pre-annotate texts too. On the other hand, there is no visual distinction between a suggestion and a user annotation. There is also no easy way to annotate two overlapping parts of a text with two separate events. Thus, testers were allowed to simplify their job and select whole sentences or even paragraphs as fragments corresponding to events.

Various existing GATE extensions and plugins were considered for our annotation experiment. Unfortunately, GATE Teamware[10] – a web-based collaborative text annotation framework which would be an obvious choice – does not currently provide good support for relation and co-reference annotation Bontcheva et al. (2013). Similarly, simple question-based user interfaces generated by the GATE Crowdsourcing plugin[11] Bontcheva et al. (2014) would not be efficient for the complex hierarchical annotation tasks tested. Thus, our experiments employed the standard GATE Developer[12] desktop interface, able to cope with the task at hand. Pre-annotations by back-end annotators were set the same way as in the other two tools. Users were instructed to perform an easier task of selecting event attributes and linking them to a reference resource first and then just selecting a text including all identified arguments and tagging it as an event.

As discussed in Section 3, it is very difficult to objectively compare semantic annotation tools from the user perspective. To minimize the danger of unfair

[7] https://code.google.com/p/wiki-links/

[8] http://knot.fit.vutbr.cz/annotations/

[9] http://rdface.aksw.org/

[10] https://gate.ac.uk/teamware

[11] https://gate.ac.uk/wiki/crowdsourcing.html

[12] https://gate.ac.uk/family/developer.html

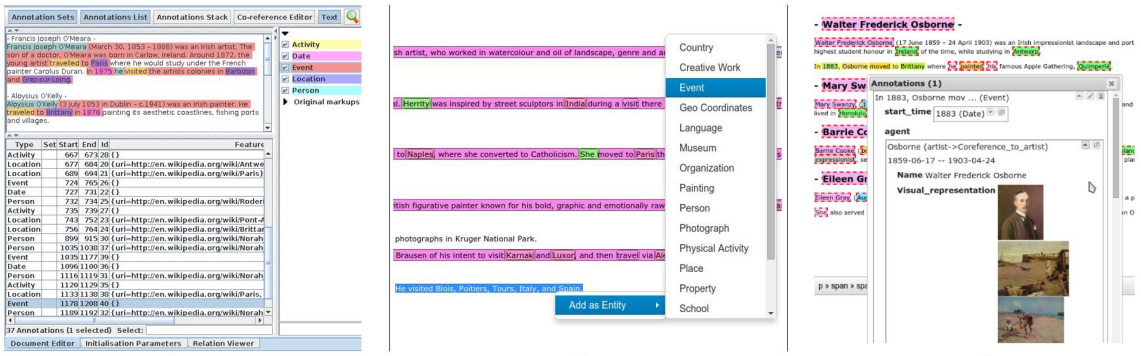

Figure 1: Event annotation screens in a) GATE, b) RDFaCE, and c) 4A.

comparison, six users that participated in the experiments had been selected to have no previous experience with neither the tools explored, nor the tasks that they used the tools for. They were 4 men and 2 women, PhD candidates or MSc. in computer science, aged 26–34. Every user spent about 20 minutes prior to the measured session familiarizing her-/himself with the tool to test while working on a specific part of the data, not included in the real testing set, yet containing all cases that appeared later during real testing (e.g., multiple values for attributes, two distinct events expressed in one sentence, suggestions that do not correspond to a correct sense, etc.) To make the comparison as fair as possible, the order in which users tested the tools was different for each user too.

Each user had about 40 minutes for annotation in each tool in the experiment. Three characteristics were collected. As summarized in Table 1, they included the amount of incorrect values entered, the number of misses – entities that were mentioned in the text but not associated with the event being annotated – and the average annotation time per event. Incorrect attributes involve all kinds of errors – incorrect selection of a textual fragment, blank or incorrect types, co-references or URLs linking entity mentions to a wrong entry in reference resources.

Table 1: Results of experiments comparing annotation tools.

tool	incorrect values	missing values	time per event
GATE	9.4 %	8.3 %	135 s
RDFaCE	8.7 %	8.8 %	193 s
4A	6.2 %	5.6 %	116 s

The overall high error rate (column “incorrect values”) can be explained by a rather strict comparison with the gold standard. For example, users were supposed to compute and enter the interval of years for an event mentioning *a woman in her 50s who travelled around* . . . Some of them entered values corresponding to 1950s.

Results correspond to the fact that the way GATE presents annotations of event attributes has often led to inconsistent results. RDFaCE was only slightly better in this respect.

A part of the problem of event slots left empty although the annotated text contains information necessary for their filling (column “missing values”) relates to pronominal references that were supposed to be linked to the referred entity. However, the difference between results of GATE and RDFaCE on one side and 4A on the other one shows that it is useful to visually distinguish system suggestions from user validated data and that 4A’s way of confirming suggestions lead to more consistent data.

Finally, the average time needed to annotate an event was higher when our testers used RDFaCE than with the other two tools. This can be explained by a rather austere user interface of the tool with a limited way to easily correct previous mistakes.

4.4 Optimizing Displayed Information

The second set of experiments explored the impact of varying amount of information presented to the user in an initial annotation view and the way users get additional information. It also asked whether users benefit from knowledge of potential alternative annotations and confidence levels of provided suggestions.

The experiments could not be done using a tool that does not allow customization and just fixes the way information is presented. We took advantage of 4A’s flexibility here and set its user interface according to required features. In particular, the setting involved limiting the set of attributes shown to users in primary disambiguation views and those that are shown when users ask for details.

The experiments particularly focused on complex entity disambiguation. As mentioned above, the data extracted from the CommonCrawl corpus was searched for links that correspond to ambiguous names of people and places in the Wikipedia. A collection of 186 excerpts used in the tests was manually verified by one of the authors. The way it had been prepared guaranteed that a random guess would lead to a 50 % error (or more, in the case of entities with more than 2 alternatives).

We primarily compared three settings of disambiguation views, differing by entity attributes shown, and looked at the impact on the speed and accuracy of the disambiguation. Users were instructed to annotate just the entity in question (highlighted in each excerpt) and choose always one of the provided suggestions. Users did not skip any disambiguation task so that we can compare just the speed and accuracy of results.

The first setting showed users an extensive list of attributes and values for the suggestion with a higher confidence. Displayed attributes involved entity type, full name, description corresponding to the first paragraph from Wikipedia or Freebase, visual representation (the first image from Wikipedia, if available), and URL. Figure 2 shows such a view. If necessary, users could follow the URL link, consult the full Wikipedia page and decide based on the full information contained there.

The second setting corresponded to the limited view some tools offer for the disambiguation task. It displayed only entity types and URLs and users were supposed to either decide based on the URL alone, or open the Wikipedia page if they felt it is necessary for the disambiguation. Note that Wikipedia URLs can help disambiguation with words in parentheses used for articles discussing entities with the same name as a primary (more famous) entity covered by the Wikipedia.

The third setting took advantage of a special disambiguation attribute that is dynamically computed from descriptions of available alternatives. It combines the disambiguation word from the Wikipedia URL and a selected part of the entity description. It is generated by a function which can be easily adapted to other data sources than Wikipedia or Freebase. The disambiguation attribute was shown together with the suggested entity type and URL so that users could again click to get more information (see Figure 3).

While the sequence of testing cases (40 for each setting) was fixed, each of 6 testers had a different order of the 3 settings (similarly as order of tools in first set of experiments). Each user had 30 minutes for each setting. Table 2 compares times and error rates

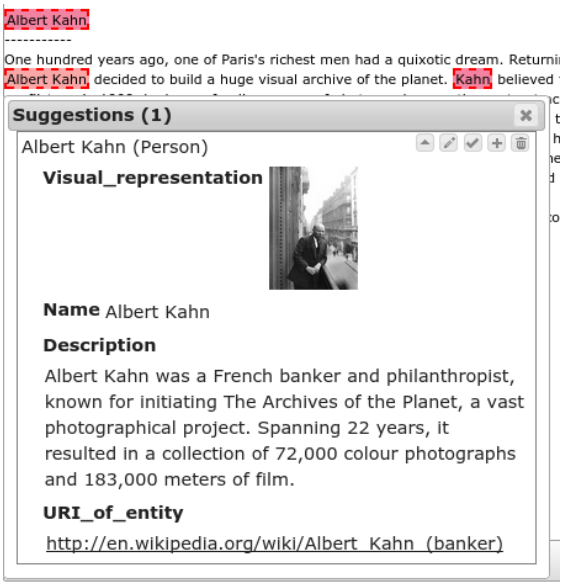

Figure 2: An example of detailed information for a suggested annotation.

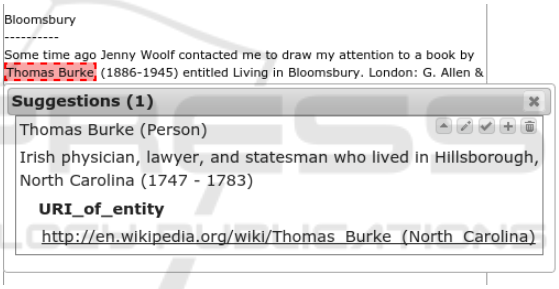

Figure 3: An example of a disambiguation attribute for a suggested annotation.

and shows how many times users clicked on the URL link to read further information.

Though there were some differences among individual testers, the overall figure (the best and the worst performing setting in terms of the average time and the error rate) remained the same for all the testers. The number of cases in which individual users consulted Wikipedia pages was always the highest in the second setting but users differed in the level they believed that seeing just a URL is enough to decide

Table 2: Results of experiments comparing three settings of the disambiguation view.

setting	average time	error rate	URL clicked
detailed information	33.92 s	6.2 %	1.3 %
only type and URL	37.26 s	27.9 %	41.7 %
disambiguation attr.	32.98 s	2.1 %	1.5 %

(which then resulted in an increased number of errors).

The relatively high number of errors is also due to the complexity of the disambiguation task. This was one of the feedback answers provided by users after all 3 sessions in a collected questionnaire. Although users tried to make as few mistakes as possible, 20+ minute sessions were felt demanding and users (not knowing how many errors they had made yet) pointed out that they could be faster if the focus would be on the speed rather than on the quality. Being confronted with the number of errors in their results, they realized the trade-off between the time and the quality and proposed context-sensitive features that would help them in particular disambiguation cases (images in the case of ambiguity between a ship name and a person, dates of deaths in the case of two persons living in different centuries, etc.).

The fact that users did not originally realize the complexity of the disambiguation task also probably explains the surprisingly high error in the case of presenting full information immediately (the first setting). Too much information that does not highlight key differences between alternatives seems to lead to a less focused work. Our future research will explore whether this can be changed when users are more experienced. On the other hand, the average time per decision and the connected low number of cases users had to consult Wikipedia pages correspond to the fact that users often skimmed full texts and images and felt they have enough information for their decisions.

The setting showing just the type and the URL proved to be the most diverse among users. Some of them opened more than 2/3 of all links and read the information on the Wikipedia page, others decided much faster but also made more errors. Although the latter could be prohibited by a penalization of errors, the second setting is clearly the worst for the task at hand. The tools that offer only this information in the disambiguation context could improve significantly by considering more informative views.

A clear winner of this part is the setting with the disambiguation attribute and the option to click on the provided URL to find details. Users made less mistakes than in other settings and the average time was the lowest. They needed to consult Wikipedia rarely. Five out of six users also indicated in the questionnaire that this setting was the most comfortable one in their eyes.

As opposed to the simplified situation prepared for the above-mentioned tested settings, the data for the next reported experiment corresponds to more realistic conditions when a name can belong to an entity that is not covered by a background knowledge base (Wikipedia, in our case) so that neither of the provided suggestions is correct. The focus on highly ambiguous names that have 5 or more alternative meanings in Wikipedia also prevents the simple selection strategy applicable in the previous settings. Users could not benefit from excluding the wrong alternative and thoughtlessly confirming the other one this time.

Two sets per 15 entity mentions were prepared while 3 of them (20 %) in each set did not correspond to neither of the alternatives. The first suggestion was correct in 9 out of 15 cases (60 %) in each set which corresponds to empirical measurement of accuracy of background suggestion service for selected texts. No threshold on the confidence level was applied so that the background suggestion service provided the best fitting alternative even in the cases when the mention did not correspond to any of the options.

One tested setting generally corresponded to the presentation of the best alternative with the disambiguation attribute shown for the most probable suggestion. Users were able to unfold and explore detailed information and visit the linked Wikipedia page.

The other setting directly showed 5 most probable suggestions and the confidence level for the best one (for technical reasons, the second to the fifth alternatives were ranked just by a global score reflecting mainly the number of visits of particular Wikipedia pages; computing real context-dependent score for each alternative would take too much time). Figure 4 shows such a case.

Three users started with the former, other three with the latter setting and continued with the other one. For the cases where none of the suggestions was correct, refusing all alternatives was taken as correct. Results are summarized in Table 3.

Table 3: One suggestion v. all alternatives.

setting	**average time**	**error rate**	**URL clicked**
one pre-selected	41.2 s	14.4 %	3.3 %
all alternatives	42.9 s	12.2 %	2.2 %

The average time of selection grew to more than 40 seconds. It reflects increased complexity of the task compared to the one in Section 4.3. It took slightly more time to consider all alternative suggestions (the last row of the table) but it resulted in a higher accuracy. Although the results are not fully conclusive and they generally depend on the quality of the suggestion-generation process and the frequency of cases where an entity is not covered by the background resource, showing alternative sugges-

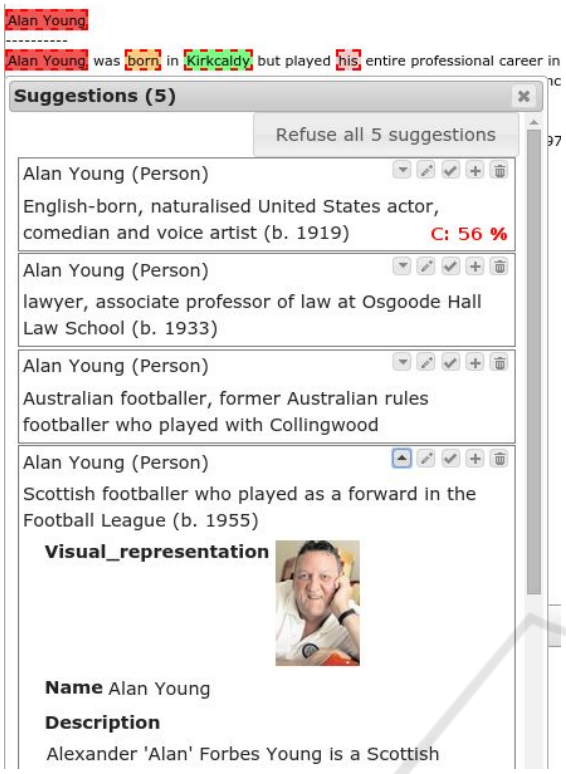

Figure 4: Alternatives shown in an initial disambiguation view.

tions seems to be the preferable option when one values quality over time. On the other hand, users' answers in the questionnaire do not indicate that knowing the confidence level of the suggestion service helped them consider less clear suggestions more carefully. Our future research will investigate this finding to a more detail.

4.5 Semantic Filtering

The third set of experiments compared two ways of annotating complex events and their attributes. It aimed at measuring what speedups can be expected when advanced semantic filtering is applied. Users were instructed to select and annotate just parts of presented paragraphs mentioning travels of a person abroad. The data was as realistic as possible – it contained sentences corresponding to other kinds of events and mentions of entities that did not play a role in the event in question.

Two sets of texts, each containing 6 events of the correct type were prepared. Users had to identify event attributes manually in the first set while the second setting involved automatic pre-annotation of attributes of each specific type and applying 4A's semantic filtering that highlights potential attribute candidates concurring to the type of the attribute being filled. The latter simplifies the role of users to assembling suggested entity annotations, adding missing ones and assembling the parts into events.

Table 4: Manually entered v. suggested annotations of entities that can fill event slots.

slot filled	incorrect values	missing values	time per event
manually	11.7 %	6.6 %	303.5 s
suggested	4.5 %	3.4 %	109.3 s

Table 4 compares the two settings. The manual process without any pre-annotation is tedious, users spend more than 5 minutes annotating one event and results contain a lot of noise. On the other hand, the 4A's semantic filtering switched on in the second setting leads to high-quality results that can be achieved relatively fast. The concept of advanced semantic filtering of suggestions will be also explored in our future work.

5 CONCLUSIONS AND FUTURE DIRECTIONS

The presented empirical study of semi-automatic textual annotation tools and their interaction components proves that user interface factors as well as involved interaction patterns have an impact on the speed of the annotation process and its results. We showed that tool developers need to pay a special attention to various aspects of the annotation interfaces, including the amount of information displayed for entity linking, the way users are notified about annotation alternatives and mechanisms of semantic templates and filtering. When done properly, the tools have a great potential to speed up computer-assisted semi-automatic annotation that is still necessary in complex relation annotation tasks.

In particular, the comparison of user interfaces and interaction patterns represented by three different annotation tools proved that the annotation consistency benefits from clear visual distinction between system suggestions and annotations validated by users. The 4A system also excelled in its comprehensible presentation of event attributes which has led to less incorrect values entered in annotations. The visually compelling interface with images representing known entities and the clear hierarchy of nested annotations allowed users to finish their tasks in shortest times using the 4A tool too.

The experiments comparing various settings of the entity disambiguation interface in the 4A tool showed

that it is beneficial to pay a special attention to the amount of information presented to users in the case of entity name ambiguity. A brief context-dependent disambiguation text supplemented by the link to a Wikipedia page or another resource providing more details helped users to make fast and accurate decisions on the entity links. 4A's intuitive semantic filtering also showed to be beneficial in terms of annotation quality and speed.

Our future work will extend the reported results towards other kinds of complex annotation tasks including data preparation for aspect-oriented sentiment analysis and annotation of textual contexts suggesting emotional states of authors. We will also support newly available entity recognition tools and frameworks, such as WAT Piccinno and Ferragina (2014) or Gerbil Röder et al. (2015), that will be employed as back-end pre-annotation components.

ACKNOWLEDGEMENTS

This work was supported by the H2020 project MixedEmotions, grant agreement No. 644632, and by the IT4IXS – IT4Innovations Excellence in Science project (LQ1602).

REFERENCES

Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., and Gorrell, G. (2013). GATE Teamware: A web-based, collaborative text annotation framework. *Lang. Resour. Eval.*, 47(4):1007–1029.

Bontcheva, K., Roberts, I., Derczynski, L., and Rout, D. (2014). The GATE Crowdsourcing Plugin: Crowdsourcing annotated corpora made easy. In *Proceedings of Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 97–100. Association for Computational Linguistics.

Ciccarese, P., Ocana, M., and Clark, T. (2012). Open semantic annotation of scientific publications using DOMEO. *Journal of Biomedical Semantics*, 3(Suppl 1).

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*. GATE.

Grassi, M., Morbidoni, C., Nucci, M., Fonda, S., and Donato, F. D. (2013). Pundit: Creating, exploring and consuming semantic annotations. In *Proceedings of the 3nd International Workshop on Semantic Digital Archives, Valletta, Malta*.

Handschuh, S., Staab, S., and Ciravegna, F. (2002). S-CREAM – Semi-automatic CREAtion of Metadata Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. In Gómez-Pérez, A. and Benjamins, V., editors, *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, volume 2473 of *Lecture Notes in Computer Science*, chapter 32, pages 165–184. Springer, Berlin, Heidelberg.

Heese, R., Luczak-Rsch, M., Paschke, A., Oldakowski, R., and Streibel, O. (2010). One click annotation. In *Proceedings of the 6th Workshop on Scripting and Development for the Semantic Web, collocated with ESWC*. Ruzica Piskac, Redaktion Sun SITE, Informatik V, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany.

Hogenboom, F., Frasincar, F., Kaymak, U., and de Jong, F. (2011). An Overview of Event Extraction from Text. *DeRiVE*.

Khalili, A., Auer, S., and Hladky, D. (2012). The RDFa Content Editor – From WYSIWYG to WYSIWYM. In *Proceedings of COMPSAC 2012 – Trustworthy Software Systems for the Digital Society*.

Kim, J., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

Maynard, D. (2008). Benchmarking textual annotation tools for the semantic web. In *6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco. European Language Resources Association (ELRA).

Maynard, D., Dasiopoulou, S., Costache, S., Eckert, K., Stuckenschmidt, H., Dzbor, M., and Handschuh, S. (2007). Knowledge web project: Deliverable D1.2.2.1.3 – Benchmarking of annotation tools.

Moro, A. and Navigli, R. (2015). SemEval-2015 Task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 288–297, Denver, Colorado.

Piccinno, F. and Ferragina, P. (2014). From TagME to WAT: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62. ACM.

Reeve, L. and Han, H. (2005). Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, pages 1634–1638, New York, NY, USA. ACM.

Röder, M., Usbeck, R., and Ngonga Ngomo, A.-C. (2015). Developing a sustainable platform for entity annotation benchmarks. In *ESWC Developers Workshop 2015*. `http://svn.aksw.org/papers/2015/ESWC_GERBIL_semdev/public.pdf`.

Smrz, P. and Dytrych, J. (2011). Towards new scholarly communication: A case study of the 4a framework. In *SePublica*, volume 721 of *CEUR Workshop Proceedings*. Ruzica Piskac, Redaktion Sun SITE, Informatik V, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany.

Smrz, P. and Dytrych, J. (2015). Advanced features of collaborative semantic annotators – the 4a system. In *Proceedings of the 28th International FLAIRS Conference*, Hollywood, Florida, USA. AAAI Press.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). 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, pages 102–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Surdeanu, M. and Heng, J. (2014). Overview of the English slot filling track at the TAC2014 knowledge base population evaluation. In *Proceedings of the TAC-KBP 2014 Workshop*.

Wang, A., Hoang, C., and Kan, M.-Y. (2013). Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation*, 47(1):9–31.

Yee, K. P. (2002). Critlink: Advanced hyperlinks enable public annotation on the web. http://zesty.ca/pubs/cscw-2002-crit.pdf.