Advanced User Interfaces for Semantic Annotation of Complex Relations in Text

Jaroslav Dytrych and Pavel Smrz

Brno University of Technology
Faculty of Information Technology
Centre of Excellence IT4Innovations
Bozetechova 2, 612 66 Brno, Czech Republic
{dytrych,smrz}@fit.vutbr.cz

Abstract. This paper deals with computer-assisted semantic annotation of text. It particularly focuses on the annotation of complex relations and linking of entities with highly ambiguous names. These tasks cannot be reliably accomplished by fully automatic methods today. Our research explores user interface features that can help the manual annotation process. We extend our original experiments published in [5] by a detailed analysis of advantages brought by the semantic filtering feature of our 4A annotation system. We also expand our user study on the annotation of highly ambiguous entities showing speed-ups brought by a presentation mode for entity candidates employing advanced disambiguation contexts.

Keywords: computer-assisted tagging; text annotation; ambiguous entity names

1 Introduction

Semantic enrichment of text forms an initial step of various text analytics techniques that have been recently applied in brand reputation management, news recommendation, market research, and many other business domains. Commercial APIs such as IBM AlchemyLanguage[1], Cogito API[2], or Ontotext S4[3] automatize the task of semantic enrichment and enable annotating key entities and basic relations between them with an acceptable degree of precision. 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 [12, 19]. The methods behind the most advanced tools usually employ machine-learning techniques which need training data. Consequently, they can be successfully applied for simple annotation tasks where the data is available but they fail for complex ones when there is not

[1] http://www.alchemyapi.com
[2] http://cogitoapi.com
[3] http://s4.ontotext.com

enough data to learn from. Manual annotation is necessary in such cases. The research presented in this paper can be seen as a quest for an optimal way to support users in the manual annotation process.

Two approaches to manual annotation can be distinguished. The first one is represented by BRAT [18] – a general linguistic annotation editor that has been used to prepare various text annotation datasets. A key characteristics of this tool is a flat structure of basic annotations and a simple way of their presentation. For example, only semantic types of particular words or multi-word expressions are shown in the case of relation annotation in BRAT and it is not easy to solve ambiguities and to link entities to an external knowledge source. The tool is suitable for highly specialised tasks such as co-reference resolution or extraction of relations between biomedical entities which have unique names. On the other hand, knowledge base linking of ambiguous people names or complex hierarchical event annotation are not supported.

The second approach to computer-assisted manual annotation employs specialised semantic editors, plugins for existing tagging tools and web browser annotation extensions [3, 6–8]. These systems are less suitable for general linguistic annotation tasks but they excel in semantic knowledge structuring (e.g., through a support of the RDF Schema) and name disambiguation. The 4A tool [16, 17] primarily used in reported experiments also belongs to this category.

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 and disambiguation of highly ambiguous entities. This was the main purpose of the set of experiments conducted by our team and reported originally in [5].

The study compared three semantic annotation tools – GATE [4], RDFaCE [9], and 4A on the task of hierarchical annotation of complex relations. 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).

This paper extends the original results in two ways. Newly conducted experiments involve more annotators and bring new insights into interaction patterns observed in complex annotation tasks employing the 4A system. Experiments exploring benefits of 4A's semantic filtering functionality involve two consecutive tasks dealing with artwork authorship mentions and artistic influences expressed in texts.

Another set of experiments searched for an optimal amount of information necessary for reliable entity disambiguation. It showed that the commonly used practice of annotation tools asking users to disambiguate entities based just on a suggested type and a displayed URL leads to a poor quality of results. Extended experiments bring new results for the case of highly ambiguous entity names

and demonstrate advantages of condensed entity views presenting task-tailored disambiguation attributes.

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 summarizes reported 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) [11]. Most of the parameters are out of scope of our study but at least some of them are covered in a publically available feature matrix that we prepared to compare usage characteristics of annotation tools from the perspective of complex annotation tasks.[4]

Maynard [10] 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 [21] 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 [14] compare semantic annotation platforms focusing on the performance of background annotation suggestion components. 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 from a historical perspective.

3 Factors Influencing Relevance of Tool Comparisons

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 of the studies.

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 individual attributes. The domain of the text being annotated (e.g., biomedical v. general)

[4] http://knot.fit.vutbr.cz/annotations/comparison.html

and its genre, register, or source (for example, news articles v. tweets) may also vary across annotation experiments. Consequently, the tasks can require particular approaches to text pre-processing and can imply different results of the automatic pre-annotation.

Datasets to be annotated do not necessarily correspond to a representative subset of relevant texts. They can focus on a chosen phenomenon and mix data accordingly. This variance can be demonstrated by differing nature of datasets prepared for previous annotation challenges. For example, the Short Text Track of the 2014 Entity Recognition and Disambiguation (ERD) Challenge[5] stressed limited contexts that naturally appear in web search queries from past TREC competitions. On the other hand, the SemEval-2015 Task 10[6] dealt with annotations relevant for sentiment analysis in microblog (Twitter) messages. The Entity Discovery and Linking (EDL) track at NIST TAC-KBP2015[7] then aimed at extracting named entity mentions, linking them to an existing Knowledge Base (KB) and clustering 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[8] – 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 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 [2] 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 times 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

[5] http://web-ngram.research.microsoft.com/erd2014/
[6] http://alt.qcri.org/semeval2015/task10/
[7] http://nlp.cs.rpi.edu/kbp/2015/index.html
[8] http://blog.commoncrawl.org

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 paid-for crowdsourcing settings that need to apply sophisticated quality control mechanisms to prevent annotators' temptation to cheat [20].

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. What quality the concept of semantic filtering brings to the annotation process.
3. To what extent the amount of information shown to disambiguators influences results.

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 system suggestions from accepted or newly entered annotations. This can have an impact on the annotation consistency.

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 generally 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 a URL link, but also full names and disambiguation contexts. The RDFaCE, on the other hand, autocompletes just URLs in this context.

The second set of experiments explores the role of semantic filtering. It is not easy to enter complex annotations, for example, interlinked hierarchical relations. Advanced mechanisms suggesting slot filling can make the process faster and more consistent. The 4A tool supports hierarchical annotation which highlights potential nested annotations if an upper-level type is known. Section 4.4 demonstrates that such an approach leads to significant speed-ups.

The last question mentioned above is covered by the final set of reported experiments. It is clear that the amount of information shown to the user and its form can influence speed and accuracy of annotation. If the displayed information is not sufficient for a decision, the user will have to search additional information. On the other hand, if a tool lets users read more than necessary, annotation speed decreases.

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 experiment too.

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.[9] First experiments 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 recognized named entities and a trigger word (verb) corresponding to artistic influences and travels and visits of people to various 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 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

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

a particular type of artistic influence relations were retrieved from the CommonCrawl data and validated by the authors. Similarly to the data for the first experiments, the dataset consists of cases in which the pre-annotation process had led to a clear consensus between the annotators. Only 20 texts mentioning several artwork influence relations were used in the study.

Final 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[10], 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 to 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.*

Extended experiments focused on highly ambiguous names that could refer to tens of entities.

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[11] 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[12] is similar to 4A in the way it annotates textual fragments and the fact it can be also deployed as a plugin for JavaScript WYSIWYG

[10] https://code.google.com/p/wiki-links/
[11] http://knot.fit.vutbr.cz/annotations/
[12] http://rdface.aksw.org/

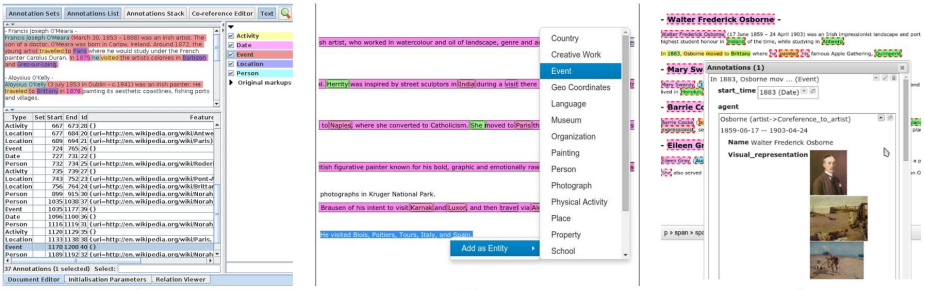

Fig. 1. Event annotation screens in a) GATE, b) RDFaCE, and c) 4A

editor TinyMCE. It can pre-annotate texts too. On the other hand, there is no visual distinction between a suggestion and a user annotation in RDFaCE. 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 the annotation experiment. Unfortunately, GATE Teamware[13] – 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 [1]. Similarly, simple question-based user interfaces generated by the GATE Crowdsourcing plugin[14] [2] would not be efficient for the complex hierarchical annotation tasks tested. Thus, our annotators used the standard GATE Developer[15] desktop interface, able to cope with the task at hand. Pre-annotations by backend 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 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 would appear 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

[13] https://gate.ac.uk/teamware
[14] https://gate.ac.uk/wiki/crowdsourcing.html
[15] https://gate.ac.uk/family/developer.html

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 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 reflect the fact that the way GATE presents annotations of event attributes often leads 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 leads 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 Effects of Semantic Filtering

Previous results showed that even though an automatic annotation process cannot identify complex relations, it is beneficial to pre-annotate entities and basic

relations and let users focus on high-level annotation tasks joining the prepared components and validating their linking to a knowledge base at the same time [5]. However, it has not been fully clear to what extent an indication of preferred types of attributes contributes to consistency of relation annotation and whether it improves annotator's comfort.

A set of 20 excerpts from documents on visual artworks (paintings, sculptures) and artistic influences were prepared for these experiments. Each of the texts mentioned several artworks, their authors and circumstances of their creation (dates, places, portrayed persons, etc.). There were also references to other artworks that inspired or influenced artists. The following paragraph shows a part of such a text:

> *Le déjeuner sur l'herbe is a large oil on canvas painting by Édouard Manet created in 1862 and 1863. Manet's composition reveals his study of the old masters, as the disposition of the main figures is derived from Marcantonio Raimondi's engraving of the Judgement of Paris (c. 1515).*

A group of 14 users identified mentions of artwork attributes in the texts first (mainly authors and creation dates). Then, they annotated influence relations between the artworks. Figure 2 demonstrates results of the process. Two configurations of the system were prepared. One highlighted potential semantic template values corresponding to the type of attribute being filled; the other one switched the semantic filtering function off. To exclude influences of the order of annotation, texts were presented to users in random order. Yet, the selection procedure guaranteed that each text will be annotated by 7 users with the semantic filtering function switched off and 7 with the function switched on.

Table 2 compares annotation results obtained with the two settings. The 4A's semantic filtering switched on led to a higher quality of results. Relative decreases of the two types of errors exceed 25 %. The annotation was also faster by 15 %. Questionnaires that the annotators had filled immediately after the experiment also revealed that 11 out of 14 users had seen the semantic filtering as a feature significantly improving their experience, the other 3 agreed that it had helped them "moderately".

Table 2. Contribution of semantic filtering

semantic filtering	**incorrect values**	**missing values**	**time per relation**
switched off	6.9 %	5.7 %	41.4 s
switched on	5.1 %	4.2 %	35.1 s

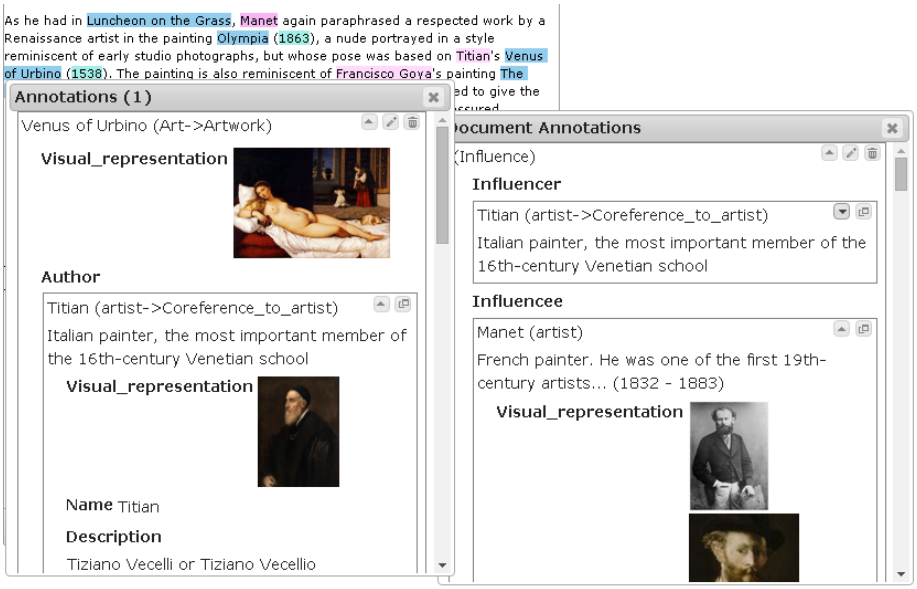

Fig. 2. Artwork attributes and influence relations

4.5 Optimizing Displayed Information

The last set of reported 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 focused on complex entity disambiguation tasks. 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 their impact on speed and accuracy of the disambiguation. Users were instructed to annotate just an entity in question (highlighted in each excerpt) and choose always one of provided suggestions. Users did not skip any disambiguation task so that we could compare just the speed and accuracy of results.

The first setting showed users an extensive list of attributes and values for suggestions with highest confidence values. 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. If necessary, users could follow the URL link, consult the relevant Wikipedia page and come to a decision 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 disambiguation words from the Wikipedia URL and a selected part of the entity description. It is generated by a function which can be easily adapted to data sources differing from Wikipedia or Freebase. The disambiguation attribute was shown together with a suggested entity type and a URL so that users could again click to get more information.

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 to the ordering of tools in the first set of experiments). Each user had 30 minutes for each setting. Table 3 compares times and error rates and shows how many times users clicked on the URL link to read further information.

Table 3. 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 %

Though there were 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 testers. The number of cases in which individual users consulted Wikipedia pages was always the highest for the second setting but users differed in the level they believed that seeing just a URL is enough to decide (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 questionnaire form. 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 so that neither of the provided suggestions is correct. The focus on highly ambiguous names that have many 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.

An experiment reported in [5] compared two settings of the disambiguation interface – one directly listing known entities sharing a name appearing in the text and the other one showing only the most probable candidate entity and letting users expand more alternatives by a click. The former showed to bring higher accuracy. That is why extended experiments covered by this paper present all alternatives to users and study how annotators perform in this case.

The number of entities sharing a given name was high (between 10 and 30) in selected texts. This corresponds to the situation when an automatic disambiguation engine has only limited information to decide so that confidence values for alternatives are small. The position of the correct choice varied in the data – 16 out of 20 texts included it in the list (on different positions) while in 4 remain-

ing cases (20 %), none of the alternatives corresponded to the entity actually mentioned in the text.

Two presentation forms of the list of alternatives were compared. The brief one listed only disambiguation contexts and enabled expanding a full list of entity attributes from the knowledge base (including description, visual representation, etc.) by a click. Figure 3 shows such a case. The expanded form listed all entity attributes directly – users had to browse through longer listings but they could easily match keywords from the text to the full entity descriptions instead.

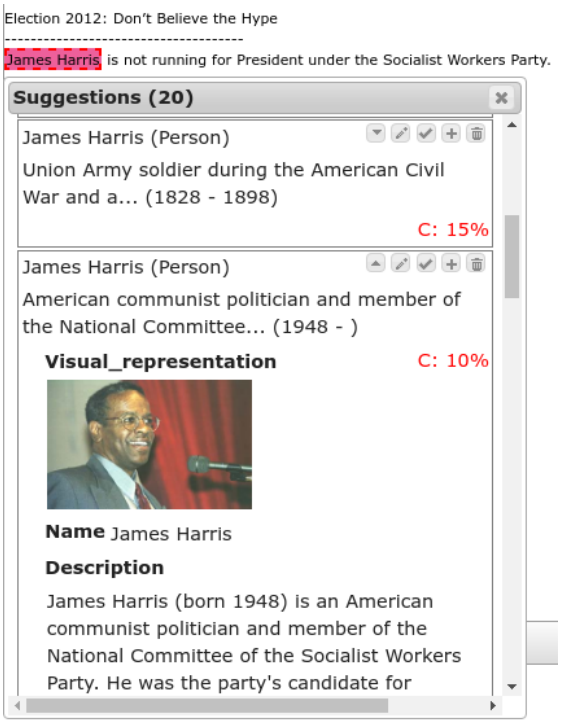

Fig. 3. An example of a full expansion of entity attributes for an alternative annotation

Each of the 20 disambiguation tasks was solved by 7 annotators using the brief view of alternatives and other 7 dealing with the expanded one. Tasks for each user were grouped by the presentation form. One half of the users started with the expanded form; the other half with the brief one.

To cope with varying numbers of alternatives as well as changing positions of the correct ones, measured total times were always divided by the rank of the correct choice. This corresponds to the most frequently observed scenario (in 86 % of the cases) in which users decided immediately when they had read an

entity description matching the annotated text. Total numbers of alternatives were considered for the 4 cases of entities not covered by the knowledge base.

Results of the experiment are summarized in Table 4. The brief view enabled users to faster scan alternatives and to select the one of their choice. This was also confirmed by post-experiment questionnaires. All but one users preferred the brief view. The relatively high error rate corresponds to the complexity of the disambiguation task. As also suggested by questionnaire answers, the accuracy would probably increase if users do not stop at the alternative that seems to fit enough and consider all others in the list. However, this could slow down the disambiguation task more than two times according to the collected data. The potential decrease of performance seems to be too high to be acceptable.

Table 4. Disambiguation with brief and expanded views on alternative entity candidates

setting	**time per alternative**	**error rate**	**clicked to expand/collapse**
brief	11.2 s	12.1 %	1.1 %
expanded	15.1 s	11.4 %	5.3 %

The last column of Table 4 characterizes interaction patterns observed when users worked with the list of alternatives. Only 1.1 % candidate entity records in the brief mode were clicked to show the full description and other attributes. The brief disambiguation attribute has been mostly seen as comprehensive. On the other hand, users dealing with the expanded view clicked the collapse button in 5.3 % of cases. The analysis of questionnaire answers then showed that this was mainly used to mark irrelevant choices. Our future research will look at these interaction patterns more closely.

5 Conclusions and Future Directions

Results of all three sets of experiments presented in this paper confirm a general finding – appropriate tool support for computer-assisted semantic annotation of text can bring significant advantages to the whole process – make it more consistent, faster and less demanding for users. If one considers a potential economic value of the manual preparation of annotation data (for example, to train advanced machine learning models for a particular complex task), it becomes critical to apply a tool with an optimal set of features for the particular annotation problem.

The empirical study focused on interaction components of semantic annotation systems suitable for complex relation annotation with highly ambiguous entities. Flexibility of the 4A system allowed us to switch on/off particular user interface components. It enabled evaluating relative contributions of specific features and showing those that bring clear benefits.

The 4A's semantic filter highlighting entities of expected types showed to be highly appreciated by users. It also led to a higher precision and slightly faster selection of attribute values. The experiments combining knowledge-based relation extraction (annotation of artwork authorship and creation dates followed by determining of artistic influences among the artworks mentioned in the text) also proved that clear distinction between system-generated annotation suggestions and user-confirmed annotations helps keep the result clear and prepare better training data for machine learning approaches.

Experiments comparing various settings of 4A's entity disambiguation interface proved that it is beneficial to pay a special attention to the amount of information presented to users in the case of entity name ambiguity and the way alternatives are presented. 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.

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 [13] or Gerbil [15], that will be employed as back-end pre-annotation components.

Acknowledgments

This work was supported by the H2020 project MixedEmotions, grant agreement No. 644632, and by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations Excellence in Science – LQ1602.

References

1. Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., Gorrell, G.: GATE Teamware: A web-based, collaborative text annotation framework. Lang. Resour. Eval. 47(4), 1007–1029 (Dec 2013), http://dx.doi.org/10.1007/s10579-013-9215-6
2. Bontcheva, K., Roberts, I., Derczynski, L., Rout, D.: 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). pp. 97–100. Association for Computational Linguistics (2014)
3. Ciccarese, P., Ocana, M., Clark, T.: Open semantic annotation of scientific publications using DOMEO. Journal of Biomedical Semantics 3(Suppl 1) (January 2012), http://www.jbiomedsem.com/content/3/S1/S1
4. 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., Peters, W.: Text Processing with GATE (Version 6). GATE (2011), http://tinyurl.com/gatebook

5. Dytrych, J., Smrz, P.: Interaction patterns in computer-assisted semantic annotation of text – an empirical evaluation. In: Proceedings of the 8th International Conference on Agents and Artificial Intelligence. SCITEPRESS, Rome, Italy (2016)
6. Grassi, M., Morbidoni, C., Nucci, M., Fonda, S., Donato, F.D.: Pundit: Creating, exploring and consuming semantic annotations. In: Proceedings of the 3nd International Workshop on Semantic Digital Archives, Valletta, Malta (2013)
7. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM – Semi-automatic CREAtion of Metadata Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. In: Gómez-Pérez, A., Benjamins, V. (eds.) Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Lecture Notes in Computer Science, vol. 2473, chap. 32, pp. 165–184. Springer, Berlin, Heidelberg (Sep 2002), http://dx.doi.org/10.1007/3-540-45810-7_32
8. Heese, R., Luczak-Rsch, M., Paschke, A., Oldakowski, R., Streibel, O.: 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 (2010)
9. Khalili, A., Auer, S., Hladky, D.: The RDFa Content Editor – From WYSIWYG to WYSIWYM. In: Proceedings of COMPSAC 2012 – Trustworthy Software Systems for the Digital Society (2012), http://svn.aksw.org/papers/2012/COMPSAC_RDFaCE/public.pdf
10. Maynard, D.: Benchmarking textual annotation tools for the semantic web. In: 6th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Marrakech, Morocco (2008), https://gate.ac.uk/sale/lrec2008/benchmarking.pdf
11. Maynard, D., Dasiopoulou, S., Costache, S., Eckert, K., Stuckenschmidt, H., Dzbor, M., Handschuh, S.: Knowledge web project: Deliverable D1.2.2.1.3 – Benchmarking of annotation tools (2007), http://knowledgeweb.semanticweb.org/semanticportal/deliverables/D1.2.2.1.3.pdf
12. Moro, A., Navigli, R.: SemEval-2015 Task 13: Multilingual all-words sense disambiguation and entity linking. In: Proceedings of the 9th International Workshop on Semantic Evaluation. pp. 288–297. Denver, Colorado (June 2015), http://www.aclweb.org/anthology/S15-2049
13. Piccinno, F., Ferragina, P.: From TagME to WAT: a new entity annotator. In: Proceedings of the first international workshop on Entity recognition & disambiguation. pp. 55–62. ACM (2014)
14. Reeve, L., Han, H.: Survey of semantic annotation platforms. In: Proceedings of the 2005 ACM Symposium on Applied Computing. pp. 1634–1638. SAC '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1066677.1067049
15. Röder, M., Usbeck, R., Ngonga Ngomo, A.C.: Developing a sustainable platform for entity annotation benchmarks. In: ESWC Developers Workshop 2015 (2015), http://svn.aksw.org/papers/2015/ESWC_GERBIL_semdev/public.pdf
16. Smrz, P., Dytrych, J.: Towards new scholarly communication: A case study of the 4a framework. In: SePublica. CEUR Workshop Proceedings, vol. 721. Ruzica Piskac, Redaktion Sun SITE, Informatik V, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany (2011)
17. Smrz, P., Dytrych, J.: Advanced features of collaborative semantic annotators – the 4a system. In: Proceedings of the 28th International FLAIRS Conference. AAAI Press, Hollywood, Florida, USA (2015)
18. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: 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. pp. 102–107. EACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012), http://dl.acm.org/citation.cfm?id=2380921.2380942
19. Surdeanu, M., Heng, J.: Overview of the English slot filling track at the TAC2014 knowledge base population evaluation. In: Proceedings of the TAC-KBP 2014 Workshop (2014)
20. Wang, A., Hoang, C., Kan, M.Y.: Perspectives on crowdsourcing annotations for natural language processing. Language Resources and Evaluation 47(1), 9–31 (2013), http://dx.doi.org/10.1007/s10579-012-9176-1
21. Yee, K.P.: Critlink: Advanced hyperlinks enable public annotation on the web (2002), http://zesty.ca/pubs/cscw-2002-crit.pdf, http://zesty.ca/pubs/cscw-2002-crit.pdf