Tools Impact on the Quality of Annotations for Chat Untangling

Jhonny Cerezo[1] , **Alexandre Bergel**[1], **Felipe Bravo-Marquez**[1,2]
[1]Department of Computer Science, University of Chile
[2]Millennium Institute for Foundational Research on Data, IMFD-Chile
jcerezo@dcc.uchile.cl, abergel@dcc.uchile.cl,
fbravo@dcc.uchile.cl

Abstract

The quality of the annotated data directly influences in the success of supervised NLP models. However, creating annotated datasets is often time-consuming and expensive. Although the annotation tool takes an important role, we know little about how it influences annotation quality. We compare the quality of annotations for the task of chat-untangling made by non-experts annotators using two different tools. The first is SLATE, an existing command-line based tool, and the second is Parlay, a new tool we developed that integrates mouse interaction and visual links. Our experimental results indicate that, while both tools perform similarly in terms of annotation quality, Parlay offers a significantly better user experience.

1 Introduction

Human linguistic annotation is essential for many natural language processing tasks. However, the construction of these datasets is extremely expensive, both in terms of annotator hours and financial cost (Snow et al., 2008). We know that the performance of many natural language processing tasks is limited by the quantity and quality of the data available to them (Banko and Brill, 2001; Snow et al., 2008). Many of the studies focus on the number of annotators and their experience in improving annotation quality. However, little has been studied about the role annotation tools play in producing quality data.

We study the effect of annotation tools in chat-untangling task. Chat-tangling occurs when simultaneous conversations arise in chat with multiple participants (Elsner and Charniak, 2008). The goal of chat-untangling is to identify the conversations in a chat-thread. In order to perform this task, annotators must maintain a complex mental representation of the ongoing conversations. Hence, a tool should minimize the task load and facilitate the annotation process. To the best of our knowledge, there is only one public dataset sufficiently large for training modern NLP architectures for this task published by Kummerfeld et al. (2019). This dataset was annotated using SLATE (Kummerfeld, 2019), a terminal based annotation tool. In SLATE, interactions are marked by keyboard commands, messages are shown as raw-text, and annotations are presented by color coding. We argue that these characteristics may influence in quality of annotations for non-experts.

This paper presents Parlay, a new annotation tool for the chat-untangling task. The main difference between Parlay and SLATE is that it integrates mouse interaction and visual links into the annotation representation. We conducted a controlled experiment with 12 non-expert participants, in which each participant was first introduced to the annotation task and then asked to use each tool to annotate 100 messages. The data to be annotated are selected from gold standard adjudicated dataset presented by Kummerfeld et al. (2019). Next, each annotation tool is evaluated in terms of usability (SUS), task load (NASA/TLX), performance and annotation time. We define annotator performance by the quality of its annotations calculated by comparing them to the gold standard (expert) annotations (Kummerfeld et al., 2019) using Cohen's Kappa coefficient (Cohen, 1960).

2 Background

Chat-untangling. Chat-untangling is an NLP task that aims to find existing conversations within a chat-log with two or more participants (Adams and Martell, 2008; Holmer, 2008; Kummerfeld, 2019; Shen et al., 2006; Elsner and Charniak, 2010, 2008). Conversations are represented by a connected graph in which the vertices are messages and the edges are relationships between them (e.g.,

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 215–220
August 5–6, 2021. ©2021 Association for Computational Linguistics

an answer to a question) (Adams and Martell, 2008; Wang et al., 2008; Kummerfeld et al., 2019; Holmer, 2008).

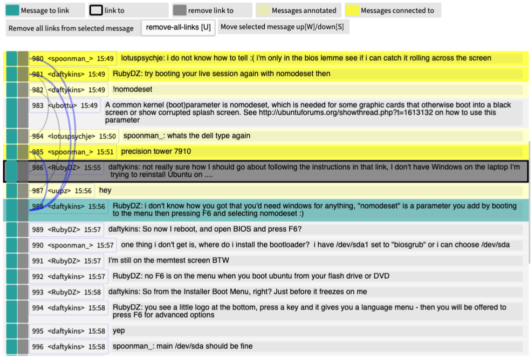
Annotation process. Manual text annotation is the process of assigning some tags to the whole text or to fragments of it. A corpus is a collection of texts on a particular topic. Annotators tag parts of the text in the corpus with labels that represent the structure or semantics of interest. The annotated data then goes through a curation process in which a curator manually resolves discrepancies and inconsistencies. Curation is primarily performed to ensure data quality (Grosman et al., 2020). One important step in curation is adjudication (Ide and Pustejovsky, 2017). In this step the curator merges the annotations from different annotators though agreement and resolves discrepancies to produce a gold standard annotation. The annotation process concludes with the release of the curated corpus to the community.

Annotation quality. To a large extent, the final quality of the annotation process depends on how human error is reduced and how discrepancies are consolidated without biasing the annotator's judgment (Chau et al., 2020). It is demonstrated that the annotation quality is related to the annotator expertise (Snow et al., 2008; Burnap et al., 2015), number of annotators (Snow et al., 2008) and the process of learning the annotation task (Teruel et al., 2018). On the other hand, annotation tools are used to assist the annotation process (e.g., file loading, user annotation interaction, data curation) (Yimam et al., 2013; Grosman et al., 2020; Kummerfeld, 2019; Yordanova et al., 2018).

Although there is evidence that user-friendly annotation tools can benefit the annotation process and reduce the annotation time (Yimam et al., 2013; Kummerfeld, 2019; Grosman et al., 2020), little is know how can they can influence to the annotator's performance. To the best of our knowledge, Grosman et al. (2020) is the only study that report changes in the inter-agreement evaluating different tools for annotation. However, there is no further understanding how this occurs at annotation time.

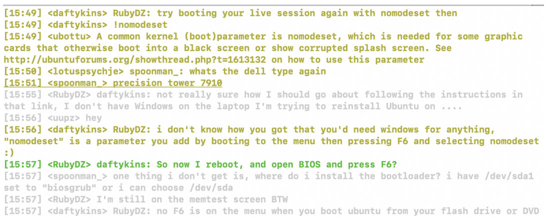
3 Methodology

In this section we introduce Parlay, a new chat-untangling annotation tool, and SLATE as baseline. Finally, we provide the evaluation methodology, criteria and our hypotheses during this study.



```
980 <spoonman_> 15:49 lotuspsychje: i do not know how to tell :( i'm only in the bios lemme see if i can catch it rolling across the screen
981 <daftykins> 15:49 RubyDZ: try booting your live session again with nomodeset then
982 <daftykins> 15:49 !nomodeset
983 <ubottu> 15:49 A common kernel (boot)parameter is nomodeset, which is needed for some graphic cards that otherwise boot into a black
                   screen or show corrupted splash screen. See http://ubuntuforums.org/showthread.php?t=1613132 on how to use this
                   parameter
984 <lotuspsychje> 15:50 spoonman_: whats the dell type again
985 <spoonman_> 15:51 precision tower 7910
986 <RubyDZ> 15:55 daftykins: not really sure how I should go about following the instructions in that link, I don't have Windows on the laptop I'm
                trying to reinstall Ubuntu on ....
987 <uupz> 15:56 hey
988 <daftykins> 15:56 RubyDZ: i don't know how you got that you'd need windows for anything, "nomodeset" is a parameter you add by booting
                   to the menu then pressing F6 and selecting nomodeset :)
989 <RubyDZ> 15:57 daftykins: So now I reboot, and open BIOS and press F6?
990 <spoonman_> 15:57 one thing i don't get is, where do i install the bootloader?  i have /dev/sda1 set to "biosgrub" or i can choose /dev/sda
991 <RubyDZ> 15:57 I'm still on the memtest screen BTW
992 <daftykins> 15:57 RubyDZ: no F6 is on the menu when you boot ubuntu from your flash drive or DVD
993 <RubyDZ> 15:58 daftykins: So from the Installer Boot Menu, right? Just before it freezes on me
994 <daftykins> 15:58 RubyDZ: you see a little logo at the bottom, press a key and it gives you a language menu - then you will be offered to
                   press F6 for advanced options
995 <daftykins> 15:58 yep
996 <daftykins> 15:58 spoonman_: main /dev/sda should be fine
```

Figure 1: Parlay. Linked messages are represented by blue arcs between messages. The gray and green messages represent the pair of messages to be connected. Annotated messages are colored in yellow.



```
[15:49] <daftykins> RubyDZ: try booting your live session again with nomodeset then
[15:49] <daftykins> !nomodeset
[15:49] <ubottu> A common kernel (boot)parameter is nomodeset, which is needed for some graphic
cards that otherwise boot into a black screen or show corrupted splash screen. See
http://ubuntuforums.org/showthread.php?t=1613132 on how to use this parameter
[15:50] <lotuspsychje> spoonman_: whats the dell type again
[15:51] <spoonman_> precision tower 7910
[15:55] <RubyDZ> daftykins: not really sure how I should go about following the instructions in
that link, I don't have Windows on the laptop I'm trying to reinstall Ubuntu on ....
[15:56] <uupz> hey
[15:56] <daftykins> RubyDZ: i don't know how you got that you'd need windows for anything,
"nomodeset" is a parameter you add by booting to the menu then pressing F6 and selecting nomodeset
:)
[15:57] <RubyDZ> daftykins: So now I reboot, and open BIOS and press F6?
[15:57] <spoonman_> one thing i don't get is, where do i install the bootloader? i have /dev/sda1
set to "biosgrub" or i can choose /dev/sda
[15:57] <RubyDZ> I'm still on the memtest screen BTW
[15:57] <daftykins> RubyDZ: no F6 is on the menu when you boot ubuntu from your flash drive or DVD
```

Figure 2: SLATE. The underlined and green messages represent the pair of messages to be linked. Messages already annotated are colored in yellow.

3.1 Annotation tools

Parlay is desktop tool developed in Pharo[1] and GT[2] whose purpose is to annotate and analyze annotations of chat-untangling task (Figure 1). Annotations can be made by linking messages with the mouse or key-commands (Table 1).

In contrast, SLATE is a terminal-based text annotation tool for different NLP tasks proposed in Kummerfeld (2019). It has the characteristics to link words, sentences and lines of text. SLATE requires each line, in the file to be annotated, to be a chat-message for chat-untangling annotation. All SLATE interactions are made by key-commands. The visual representation of messages is simple plain text with no distinctive format for its components (Figure 2). The full list of functional steps to annotate in SLATE is given in Table 1.

Parlay and SLATE differ on two essential aspects. The first is the linking visualization, for which Parlay creates a visual-arc in blue color, and

[1] Pharo is a pure object-oriented programming language and offers a flexible programming environment (Bergel et al., 2013).

[2] GT, which stands for Glamorous Toolkit, is a "moldable programming environment", https://gtoolkit.com.

Tool	SLATE	Parlay
Select message	Move messages by pressing arrow key one line at the time to select a message.	Scroll down or up, then left-click on a message's green button at the left part of each message to select it.
Create annotation	Move to select both messages then press key "D" to link them.	Select the first message, then scroll down/up and left-click in a message's username to link it to the first one.
Remove annotation	Select a message, then press key "U". It eliminates all links from the selected messages.	i) Select a message, then press key "R". It eliminates all links from the selected message. ii) Select the first message, then scroll down/up and left-click on the gray button at the left of the message. It eliminates only the link between those messages.
Validate annotation	Select a message that has already been annotated. This action will change the color of messages that are linked to the selected message.	i) Select a message that has already been annotated. ii) Hover over the left green button of a message. Both options will highlight the color of links and messages that are linked to the selected message.

Table 1: Functional differences between Parlay and SLATE.

		First session		Second session	
	Nro	**File**	**Tool**	**File**	**Tool**
Setting A	6	2015-08-10	SLATE	2015-10-19	Parlay
Setting B	6	2015-08-10	Parlay	2015-10-19	SLATE

Table 2: Experiment settings.

SLATE shows a change in the color of messages. Second, Parlay allows one to use the mouse to define annotations, whereas in SLATE the annotations are made by keyboard commands. At first, these two differences may look superficial. However, it is reasonable to think that they may significantly impact the user experience.

3.2 Evaluation Design

This study aims to assess the usability, task load, performance and annotation time of the SLATE and Parlay annotation tools. In order to create fair conditions for evaluating both tools, we conducted a controlled experiment.

Participants. We gathered 12 participants among university graduates (4), master students (4), master graduate (1), doctoral students (1), postdoctoral (2). None of the participants had a background in text annotation. All of them were experienced in the use of Linux, and none of them were native English speakers, although all of them declared to have a proficient level of English.

Data. We selected two files from Kummerfeld et al. (2019) adjudicated dataset that was developed to calculate inter-agreement. The adjudicated file is considered as the gold standard annotation to which we compare our non-expert participants' annotations. Then, we selected a pair of files with high similarity by two criteria: a) the agreement between annotators against the adjudicated file, and b) the number of annotated conversations.

Design. The controlled experiment was divided in three sessions, one introductory and two evaluation sessions. In the introductory session each participant had to i) answer a demographic questionnaire, ii) read an introductory document to the chat-untangling task, iii) read the annotation guidelines developed by Kummerfeld et al. (2019), and iv) answer a chat-untangling annotation exercise. The exercise consists of annotating three 5-messages long conversations without the help of Parlay or SLATE. The moderator then discuses the participants' annotations. The annotation exercise ensures the participants' maximum understanding of the chat-untangling task.

In the consequent evaluation sessions the participants had to i) annotate a file using one of the two tools, and ii) answer questionnaires. The tool for each session is randomly selected, whereas the annotated file is always the same for each evaluation session. We name each file-tool combination as an *experiment setting*. Such that we get two experiment settings, A and B. Where each *experiment setting* considers 6 participants and the participants use both tools in turn for the annotation task. For instance, a participant in setting A uses SLATE in the first evaluation session and Parlay in the second. In Table 2 we present the two experiment settings in detail.

Validation Criteria. Parlay and SLATE are evaluated in terms of usability, task load, annotation time and performance. The usability is measured by the System Usability Scale (SUS) (Brooke, 1996). The task load is measured using NASA/TLX (Hart and Staveland, 1988). The annotation time is determined from the annotation-logs. Lastly, the performance or quality of the annotation is measured by calculating the Cohen's Kappa (Cohen, 1960) for agreement between each participant's annotations and the expert adjudicated annotations (gold standard) presented in Kummerfeld et al. (2019) work. We say that the higher the Kappa score (κ) the better the performance.

	Parlay		SLATE	
	Mean	SD	Mean	SD
Mental	5.833	1.992	6.167	2.167
Physical	3.417	2.151	4	2.216
Temporal	4.417	2.429	4.25	2.34
Performance	5.5	2.908	5.167	1.946
Effort	5.25	1.815	6	2.174
Frustration	3.667	1.969	5.583	0.832

Table 3: Comparison of average and standard deviation of NASA/TLX scores between Parlay and SLATE.

Hypotheses. To analyze this study results, we performed some statistical tests to verify the following alternative hypotheses:

- H1: The participants' performance is affected by the annotation tool.
- H2: The participants' performance is affected by the experience gained in the annotation sessions.

We evaluated the significance of the performance *kappa* using the statistical t-test (De Winter, 2013) for small sample size (De Winter, 2013). We reject the null hypotheses if $p\text{-}value < 0.05$.

4 Results

In this section we present the results of our methodology of evaluation for i) the task load, ii) usability, iii) performance and iv) annotation time.

Task load. Table 3 shows the averaged results of participants' answers for the NASA/TLX dimensions. On average mental demand, physical demand, effort and frustration are lower in Parlay, whereas SLATE shows less temporal demand, although the difference is slight. Lastly, the users report that they performed better using Parlay. Overall Parlay reports less task load with the exception of being slightly more temporal demanding.

Usability Table 4 shows the mean and standard deviation values of SUS scores with the two annotation tools. If we pay attention to questions regarding positive (i.e., Q1, Q3, Q5, Q7 and Q9) and negative (i.e., Q2, Q4, Q6, Q8 and Q10) aspects of usability we find that: i) SLATE has similar average scores in positive and negative; and ii) Parlay rates higher to questions regarding the positive and lower in negative aspects of usability. A closer look at the ten component SUS scores we can see that Parlay is perceived more usable by the participants. This suggests that Parlay achieved better usability compared to SLATE.

	Parlay		SLATE	
	Mean	SD	Mean	SD
Q1: Willing to use the system	5.083	2.644	3	1.859
Q2: Complexity of the system	3.583	2.429	5.167	2.329
Q3: Ease of use	7.333	2.348	6.167	2.25
Q4: Need of support to use	3.833	2.443	5.25	2.491
Q5: Integrity of Functions	7.5	0.431	5.833	2.823
Q6: Inconsistency	3.25	2.454	3.75	1.658
Q7: Intuitiveness	7.333	2.309	5.333	2.498
Q8: Cumbersomeness to use	3.583	2.575	5.417	2.353
Q9: Feeling confident to use	7.667	1.67	5.667	2.348
Q10: Required learning-effort	4.083	2.353	4.667	2.309
Positive (odd)	6.983	2.425	5.2	2.563
Negative (even)	3.667	2.384	4.85	2.254

Table 4: Comparison of average and standard deviation of SUS scores between Parlay and SLATE.

		Sample	Time (min)		Performance	
		Size	Mean	SD	Mean	SD
Tool	**Parlay**	12	48.5	21.211	0.666	0.079
	SLATE	12	41.083	16.714	0.606	0.12
Session	**1st**	12	50.167	15.643	0.582	0.096
	2nd	12	39.417	21.249	0.69	0.084

Table 5: Performance (κ) and annotation time (min) results by session of annotation and annotation tool.

Annotation time. SLATE reported less annotation time in minutes in our participants (Mean=41.083, SD=16.714) compared to Parlay (Mean=48.5, SD=21.211). Lastly the second evaluation session reported less annotation time in our participants (Mean=39.417, SD=21.249) compared to Parlay (Mean=50.167, SD=15.643).

Performance. We assess our hypotheses by calculating the performance (κ) by two criteria: i) the tools used in the annotation task (H1) and ii) the session in which the participants annotate (H2). For H1 there was a non-significant difference in the scores for Parlay and SLATE with p-value=0.1583. For H2 there was a significant difference in the scores for the first evaluation session and second evaluation session with p-value=0.007636. Therefore, the quality of annotations increases as annotators gain experience. Finally, the annotation tool used does not influence the quality of annotations.

5 Conclusion

In this paper we evaluate the influence of the annotation tool for non-expert users in terms of data quality and user experience in the chat-untangling task. To achieve our purpose we introduce a new annotation tool named Parlay and establish SLATE as our baseline. Subsequently, we conducted a controlled experiment with 12 non-expert annotators in which each participant annotated 100 messages on each tool in turn. Each tool was evaluated under i) usability, ii) task load, iii) annotation time and iv) annotation performance. Lastly, we establish that the quality of annotations is measured by calculating the agreement (κ) between participants' annotations and the gold annotated data (Kummerfeld et al., 2019).

The results indicate that the tools did not show significant differences in the annotators' performance outcome on the chat-untangling task. On the other hand, participants showed better performance in the second session, presumably due to a gain in experience. The annotation time is also lower in SLATE. Complementary to these results, we also report that Parlay scored better in usability and task load. Where participants highlighted that i) link representation between annotated messages and ii) mouse interaction where the main characteristics that made Parlay. In conclusion, Parlay offers a better user experience while achieving comparable annotation performance.

The study of annotation quality is an important issue for future development of NLP models. Tools remain as an important factor in the development of high-quality training datasets for NLP tasks (Grosman et al., 2020). Despite the results presented in this study, further work is required to get a thorough understanding of how the annotation tool affects the quality of chat-untangling data.

This study contributes to the area by introducing Parlay, a new tool for the chat-untangling task. We believe that an important contribution of our work is the methodology we propose, which allows us to compare annotation tools according to both data quality and user experience. For future work we aim to study the inter-annotator agreement of each tool. As well as, the discrepancies between expert an non-expert annotations.

Acknowledgements

We thank Renato Cerro for his review and editing and we thank 3 "anonymous" reviewers for their so-called insights. Felipe Bravo-Marquez was funded by ANID FONDECYT grant 11200290, U-Inicia VID Project UI-004/20 and ANID - Millennium Science Initiative Program - Code ICN17_002. Alexandre Bergel thanks Lam Research and the ANID FONDECYT Regular 1200067 for partially sponsoring the work presented in this paper.

References

Paige H Adams and Craig H Martell. 2008. Topic detection and extraction in chat. In *2008 IEEE international conference on Semantic computing*, pages 581–588. IEEE.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 26–33.

Alexandre Bergel, Damien Cassou, Stéphane Ducasse, and Jannik Laval. 2013. *Deep Into Pharo*. Square Bracket Associates.

John Brooke. 1996. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, page 189.

Alex Burnap, Yi Ren, Richard Gerth, Giannis Papazoglou, Richard Gonzalez, and Panos Y Papalambros. 2015. When crowdsourcing fails: A study of expertise on crowdsourced design evaluation. *Journal of Mechanical Design*, 137(3).

Hung Chau, Saeid Balaneshin, Kai Liu, and Ondrej Linda. 2020. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 74–86.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Joost CF De Winter. 2013. Using the student's t-test with extremely small sample sizes. *Practical Assessment, Research, and Evaluation*, 18(1):10.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842.

Micha Elsner and Eugene Charniak. 2010. Disentangling chat. *Computational Linguistics*, 36(3):389–409.

Jonatas S Grosman, Pedro HT Furtado, Ariane MB Rodrigues, Guilherme G Schardong, Simone DJ Barbosa, and Hélio CV Lopes. 2020. Eras: Improving the quality control in the annotation process for natural language processing tasks. *Information Systems*, page 101553.

Sandra G Hart and Lowell E Staveland. 1988. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier.

Torsten Holmer. 2008. Discourse structure analysis of chat communication. *Language@ Internet*, 5(10).

Nancy Ide and James Pustejovsky. 2017. *Handbook of linguistic annotation*. Springer.

Jonathan K Kummerfeld. 2019. Slate: A super-lightweight annotation tool for experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12.

Jonathan K Kummerfeld, Sai R Gouravajhala, Joseph J Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42.

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. 2008. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263.

Milagro Teruel, Cristian Cardellino, Fernando Cardellino, Laura Alonso Alemany, and Serena Villata. 2018. Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Yi-Chia Wang, Mahesh Joshi, William W Cohen, and Carolyn Penstein Rosé. 2008. Recovering implicit thread structure in newsgroup style conversations. In *ICWSM*.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6.

Kristina Yordanova, Adeline Paiement, Max Schröder, Emma Tonkin, Przemyslaw Woznowski, Carl Magnus Olsson, Joseph Rafferty, and Timo Sztyler. 2018. Challenges in annotation of user data for ubiquitous systems: Results from the 1st arduous workshop.