User Performance Indicators In Task-Based Data Collection Systems

Jon Chamberlain, Cliff O'Reilly
School of Computer Science and Electronic Engineering
University of Essex
Wivenhoe Park, CO4 3SQ England
{jchamb,coreila}@essex.ac.uk

Abstract

When attempting to analyse and improve a system interface it is often the performance of system users that measures the success of different iterations of design. This paper investigates the importance of sensory and cognitive stages in human data processing, using data collected from Phrase Detectives, a text-based game for collecting language data, and discusses its application for interface design.

1 Introduction

When attempting to analyse and improve a system interface it is often the performance of system users that measures the success of different iterations of design. The metric of performance depends on the context of the task and what is considered the most important outputs by the system owners, for example one system may desire high quality output from users, whereas another might want fast output from users [RC10].

When quality is the performance measure it is essential to have a trusted gold standard with which to judge the user's responses. A common problem for natural language processing applications, such as co-reference resolution, is that there is not sufficient resources available and creating them is both time-consuming and costly [PCK+13].

Using user response time as a performance indicator presents a different set of problems and it may not necessarily be assumed that speed correlates to quality. A fast response may indicate a highly trained user responding to a simple task and conversely a slow response might indicate a difficult task that requires more thought.

Figure 1: Stages of processing in human cognition.

It is therefore important to understand what is happening to the user during the response and whether there is anything that can be done to the system to improve performance.

This paper investigates the importance of sensory and cognitive stages in human data processing, using data collected from Phrase Detectives, a text-based game for collecting language data, and attemps to isolate the effect of each stage. Furthermore we discuss the implications and its application for interface design.

2 Related Work

The analysis of timed decision making has been a key experimental model in Cognitive Psychology. Studies in Reaction (or Response) Time (RT) show that the human interaction with a system can be divided into discrete stages: incoming stimulus; mental response; and behavioural response [Ste69]. Although traditional psychological theories follow this model of progression from perception to action, recent studies are moving more towards models of increasing complexity [HMU08].

For our investigation we distinguish between 3 stages of processing required from the user to elicit an output response from input stimuli (see also Figure 1):

In: U. Kruschwitz, F. Hopfgartner and C. Gurrin (eds.): Proceedings of the MindTheGap'14 Workshop, Berlin, Germany, 4-March-2014, published at http://ceur-ws.org

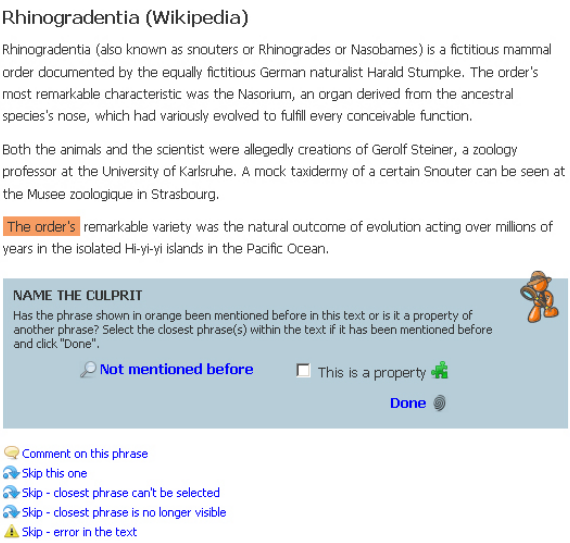

Figure 2: A task presented in Annotation Mode.

1. input processing (sensory processing) where the user reads the text and comprehends it;
2. decision making (cognitive processing) where the user makes a choice about how to complete the task;
3. taking action (motor response) to enter the response into the system interface (typically using a keyboard or mouse).

This model demonstrates how a user responds to a task and can be seen in many examples of user interaction in task-based data collection systems. In crowdsourcing systems, such as Amazon's Mechanical Turk[1] or data collection games, a user is given an input (typically a section of text or an image) and asked to complete a task using that input, such as to identify a linguistic feature in the text or to categorise objects in an image [KCS08]. The model can also be seen in security applications such as reCAPTCHA, where the response of the user proves they are human and not an automated machine [vAMM+08]. As a final example, the model can be seen in users' responses to a search results page, with the list of results being the input and the click to the target document being the response [MTO12].

The relationship between accuracy in completing a task and the time taken is known as the Speed Accuracy Trade-off. Evidence from studies in ecological decision-making show clear indications that difficult tasks can be guessed where the costs of error are low. This results in lower accuracy but faster completion time [CSR09, KBM06]. Whilst studies using RT as a measure of performance are common, it has yet to be incorporated into more sophisticated models predicting data quality from user behaviour [RYZ+10, WRfW+09, KHH12, MRZ05].

[1] http://www.mturk.com

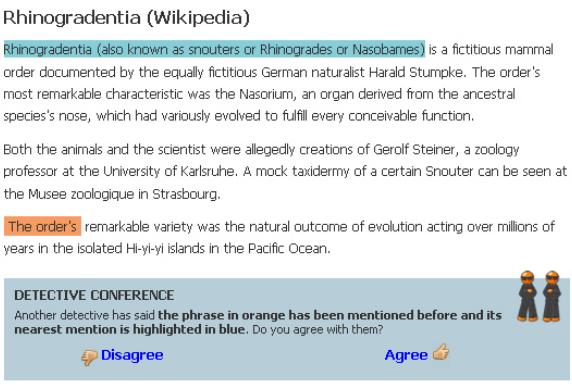

Figure 3: A task presented in Validation Mode.

3 Data Collection

Phrase Detectives is a game-with-a-purpose designed to collect data on anaphoric co-reference[2] in English documents [CPKs08, PCK+13].

The game uses 2 modes for players to complete a linguistic task. Initially text is presented in Annotation Mode (called Name the Culprit in the game - see Figure 2) where the player makes an annotation decision about a highlighted markable (section of text). If different players enter different interpretations for a markable then each interpretation is presented to more players in Validation Mode (called Detectives Conference in the game - see Figure 3). The players in Validation Mode have to agree or disagree with the interpretation.

The game was released as 2 interfaces: in 2008 as an independent website system (PD)[3] and in 2011 as an embedded game within the social network Facebook (PDFB).[4] Both versions of the Phrase Detectives game were built primarily in PHP, HTML, CSS and JavaScript, employ the same overall game architecture and run simultaneously on the same corpus of documents.

One of the differences between Phrase Detectives and other data collection games is that it uses preprocessing to offer the players a restricted choice of options. In Annotation Mode the text has embedded code that shows all selectable markables; In Validation Mode the player is offered a binary choice of agree-

[2] Anaphoric co-reference is a type of linguistic reference where one expression depends on another referential element. An example would be the relation between the entity *'Jon'* and the pronoun *'his'* in the text *'Jon rode his bike to school'*.

[3] http://www.phrasedetectives.com

[4] https://apps.facebook.com/phrasedetectives

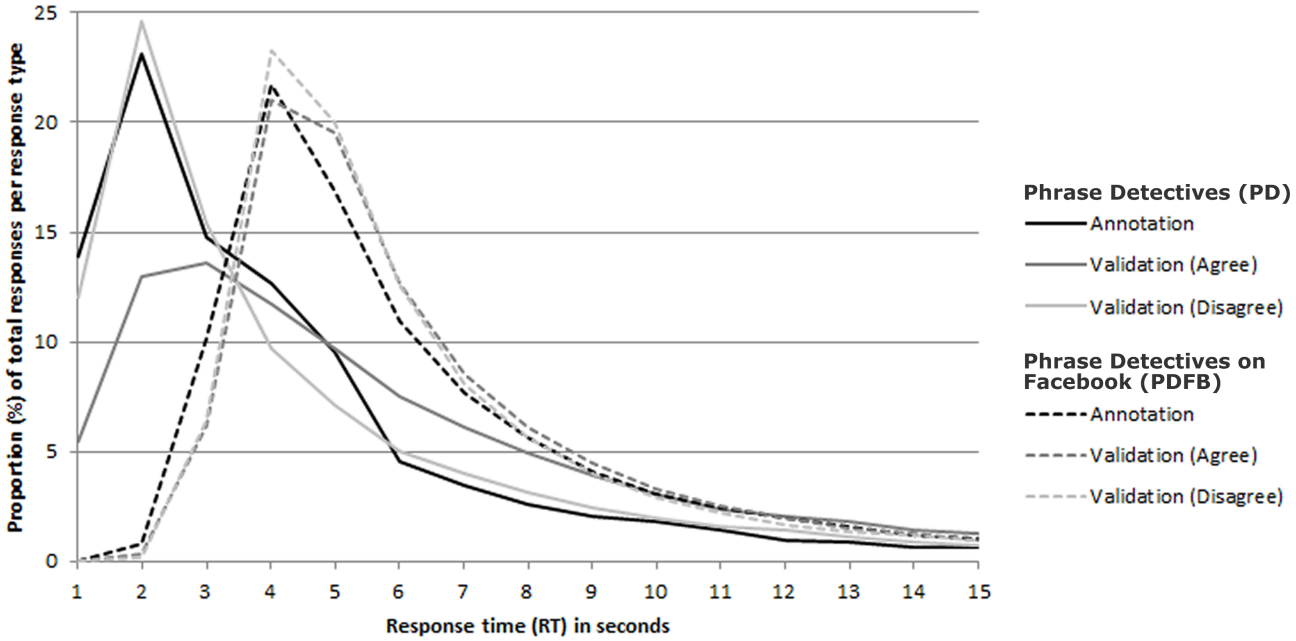

Figure 4: Proportional frequency of RT in the 2 modes of the 2 interfaces of Phrase Detectives.

Table 1: Total responses for the 2 modes in the 2 interfaces of Phrase Detectives.

	PD	PDFB
Total Annotations	1,096,575	520,434
Total Validations (Agree)	123,224	115,280
Total Validations (Disagree)	278,896	199,197

ing or disagreeing with an interpretation. This makes the interface more game-like and allows the data to be analysed in a more straightforward way as all responses are clicks rather than keyboard typing. In this sense it makes the findings more comparable to search result tasks than reCAPTCHA typing tasks.

4 Analysis

In order to investigate the human data processing in the Phrase Detectives game the RT was analysed in different ways. All data analysed in this paper is from the first 2 years of data collection from each interface and does not include data from markables that are flagged as ignored.[5] Responses of 0 seconds were not included because they were more likely to indicate a problem with the system rather than a sub 0.5 second response. Responses over 512 seconds (8:32 minutes)[6] were also not included and outliers do not represent more than 0.5% of the total responses.

An overview of the total responses from each interface shows the PDFB interface had proportionately

[5]System administrators manually correct pre-processing errors by tagging redundant markables to be ignored.

[6]The upper time limit is set at 512 seconds because the data is part of a larger investigation that used RT grouped by a power function and it is assumed no task would take longer than this.

Table 2: Minimum, median and mean RT from a random sample of 50,000 responses of each response type from PD and PDFB.

	PD	PDFB
Annotation RT (min)	1.0s	2.0s
Annotation RT (med)	3.0s	6.0s
Annotation RT (mean)	7.2s	10.2s
Validation (Agr) RT (min)	1.0s	1.0s
Validation (Agr) RT (med)	5.0s	6.0s
Validation (Agr) RT (max)	10.0s	10.5s
Validation (Dis) RT (min)	1.0s	2.0s
Validation (Dis) RT (med)	3.0s	6.0s
Validation (Dis) RT (mean)	8.4s	9.9s

fewer annotations to validations than the PD interface indicating that the players in the latter disagreed with each other more (see Table 1). A random sample of 50,000 responses per response type (annotation, agreeing validation, and disagreeing validation) shows that users respond differently between the 2 interfaces (see Table 2). The data was also plotted as a proportional frequency of RT, with a focus on the first 15 seconds (see Figure 4).

There is a significant difference in the RT between interfaces ($p<0.05$, unpaired t-test). This may indicate a higher level of cheating and spam in PD however PDFB may be slower because it had to load the Facebook wrapper in addition to the interface. This is supported by the minimum RT for PDFB being 2.0s in Annotation and Validation (Disagree) Modes, where it could be assumed that this is the system's maximum speed. The 2 interfaces differ in the proportion of responses 2 seconds or less (almost a third of all

responses in PD but a negligible amount in PDFB). One of the motivations for this research is to understand the threshold where responses can be excluded based on predicted RT rather than comparison to a gold standard.

The RT for validations was slower than for annotations in the PD interface. This is counter-intuitive as Annotation Mode has more options for the user to choose from and requires a more complex motor response. One of the assumptions in the orginal game design was that a Validation Mode would be faster than an Annotation Mode and it would make data collection more efficient.

The data was further analysed to investigate the 3 stages of user processing. Different data models were used to isolate the effect of the stage in question and negate the influence of the 2 other stages.

4.1 Input processing

A random sample of 100,000 validation (agree and disagree) responses were taken from the PDFB corpus. The RT and character distance at the start of the markable were tested for a linear correlation, the hypothesis being that more input data (i.e., a long text) will require more time for the player to read and comprehend. Validation Mode was used because it always displays the same number of choices to the player no matter what the length of the text (i.e., 2) so the action and decision making stages should be constant and any difference observed in RT would be due to input processing.

There was a significant correlation between RT and the amount of text displayed on the screen ($p<0.05$, Pearson's Correlation) which supports the hypothesis that processing a larger input takes longer time.

4.2 Decision making

The decision making stage was investigated using an analysis of 5 documents in the PD corpus that had a double gold standard (i.e., had been marked by 2 language experts), excluding markables that were ambiguous (i.e., the 2 experts did not agree on the best answer) or where there was complete consensus. The comparison of paired responses of individual markables minimises the effect of processing time and the action time is assumed to be evenly distributed.

The analysis shows that an incorrect response takes longer, significantly so in the case of making an annotation or agreeing with an annotation ($p<0.05$, paired t-test) - see Table 3. Given that this dataset is from PD where there are a high number of fast spam responses it is feasible that the true incorrect RT is higher. Taking longer to make an incorrect response is indicative of a user who does not have a good understanding of the task or that the task is more difficult than usual.

Table 3: Mean RT for aggregated correct and incorrect responses in the 2 modes from 122 gold standard markable observations (80 in the case of Validation Disagree). * indicates $p<0.05$.

	Correct	**Incorrect**
Annotation*	10.1s	12.8s
Validation (Agree)*	13.5s	17.7s
Validation (Disagree)	14.5s	15.0s

Table 4: Minimum, median and maximum RT for clicking actions in Annotation Mode from 6,176 markables ($p<0.01$).

	Min	**Med**	**Max**
1 click (DN, NR)	1.0s	5.0s	123.3s
2 clicks (DO1)	1.0s	9.8s	293.0s
3 clicks (DO2, PR1)	2.0s	12.0s	509.0s

Mean RT is slower than the general dataset (Table 2). One explaination is that the gold standard was created from some of the first documents to be completed and the user base at that time would mostly have been interested early adopters, beta testers and colleagues of the developers rather than the more general crowd that developed over time, including spammers making fast responses.

4.3 Taking action

A random sample of 100,000 markables and associated annotations was taken from completed documents from both interfaces where the markable starting character was greater than 1,000 characters. Annotations were grouped on the minimum number of clicks that would be required to make the response (any markables that had no responses in any group were excluded). Thus the effect of input processing speed was minimised in selected markables and decision making time is assumed to be evenly distributed.

- 1 click response, including Discourse-New (DN) and Non-Referring (NR);
- 2 click response, including Discourse-Old (DO) where 1 antecedent was chosen;
- 3 click response, including DO where 2 antecedents were chosen and Property (PR) where 1 antecedent was chosen.

There is a significant difference between each group ($p<0.01$, paired t-test), implying that the motor response per click is between 2 to 4 seconds, although for some tasks it is clearly faster as can be seen in the minimum RT. This makes the filtering of responses

below a threshold RT important as in some cases the user not would have enough time to process the input, make a decision and take action. This will be dependent of how difficult the task is to repond to.

Here the actions require the user to click on a link or button but this methodology can be extended to cover different styles of input, for example freetext entry. Freetext is a more complicated response because the same decision can be expressed in different ways and automatic text processing and normalisation would be required. However, when a complex answer might be advantageous, it is usefulto have an unrestricted way of collecting data allowing novel answers to be recorded. To this end the Phrase Detectives game allowed free-text comments to be added to markables.

5 Discussion

By understanding the way users interact with a system each task response time can be predicted. In the case of the Phrase Detectives game we can use a prediction of what the user should do for a given size of input to process, task difficulty and data entry mode. The same could be applied to any task driven system such as search, where the system returns a set of results from a query of known complexity with a set of actionable areas that allow a response to be predicted even when the user is unknown.

When the system is able to predict a response time for a given input, task and interface combination user performance can be measured, with users that perform as predicted being used as a pseudo-gold standard so the system can learn from new data. Outlier data can be filtered; a response that is too fast may indicate the user is clicking randomly or that it is an automated or spam response; a response that is too slow may indicate the user is distracted, fatigued or does not understand the task and therefore the quality of their judgement is likely to be poor.

The significant results uncovered by the analysis of the Phrase Detectives data should be treated with some caution. Independent analysis of each processing stage is not entirely possible for log data because users are capable of performing each stage simultaneously, i.e., by making decisions and following the text with the mouse cursor whilst reading the text. A more precise model could be achieved with eye-tracking and GOMS (Goals, Operators, Methods, and Selection) rule modelling [CNM83] using a test group to establish baselines for comparison to the log data or by using implicit user feedback from more detailed logs [ABD06]. Without using more precise measures of response time this method is most usefully employed as a way to detect and filter spam and very poor responses, rather than as a way to evaluate and predict user performance.

Modelling the system and measuring user performance allows designers to benchmark proposed changes to see if they have the desired effect, either an improvement in user performance or a negligible detriment when, for example, monetising an interface by adding more advertising. Sensory and motor actions in the system can be improved by changes to the interface, for example in the case of search results, ensuring the results list page contains enough data so the user is likely to find their target but not so much that it slows the user down with input processing. Even simple changes such as increasing the contrast or size of the text might allow faster processing of the input text and hence improve user performance. Decision making can be improved through user training, either explicitly with instructions and training examples or implicitly by following interface design conventions so the user is pre-trained in how the system will work.

Predicting a user response is an imprecise science and other human factors should be considered as potentially overriding factors in any analysis. A user's expectations of how an interface should operate combined with factors beyond measurement may negate careful design efforts.

6 Conclusion

Our investigation has shown that all three stages of user interaction within task-based data collection systems (processing the input; making a decision; and taking action) have a significant effect on the response time of users and this has an impact on how interface design elements should be applied. Using response time to evaluate users from log data may only be accurate enough to filter outliers rather than predict performance, however this is the subject of future research.

6.0.1 Acknowledgements

The authors would like to thank the reviewers and Dr Udo Kruschwitz for their comments and suggestions. The creation of the original game was funded by EPSRC project AnaWiki, EP/F00575X/1.

References

[ABD06] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 19–26, New York, NY, USA, 2006. ACM.

[CNM83] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983.

[CPKs08] Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. Phrase Detectives: A web-based collaborative annotation game. In *Proceedings of the International Conference on Semantic Systems (I-Semantics'08)*, 2008.

[CSR09] Lars Chittka, Peter Skorupski, and Nigel E Raine. Speed–accuracy tradeoffs in animal decision making. *Trends in Ecology & Evolution*, 24(7):400–407, 2009.

[HMU08] Hauke R. Heekeren, Sean Marrett, and Leslie G. Ungerleider. The neural systems that mediate human perceptual decision making. *Nature reviews. Neuroscience*, 9(6):467–479, June 2008.

[KBM06] Leslie M Kay, Jennifer Beshel, and Claire Martin. When good enough is best. *Neuron*, 51(3):277–278, 2006.

[KCS08] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[KHH12] Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 467–474, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[MRZ05] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The Peer-Prediction method. *Management Science*, 51(9):1359–1373, September 2005.

[MTO12] Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. Learning to predict response times for online query scheduling. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 621–630, New York, NY, USA, 2012. ACM.

[PCK+13] Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation. *ACM Transactions on Interactive Intelligent Systems*, 2013.

[RC10] Filip Radlinski and Nick Craswell. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674. ACM, 2010.

[RYZ+10] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, August 2010.

[Ste69] Saul Sternberg. The discovery of processing stages: Extensions of Donders' method. *Acta Psychologica*, 30:276–315, 1969.

[vAMM+08] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[WRfW+09] Jacob Whitehill, Paul Ruvolo, Ting fan Wu, Jacob Bergsma, and Javier Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, page 2035–2043, December 2009.