A Task Completion Engine to Enhance Search Session Support for Air Traffic Work Tasks

Yashar Moshfeghi[1], Raoul Rothfeld[1], Leif Azzopardi[2], Peter Triantafillou[1]
`Yashar.Moshfeghi@glasgow.ac.uk, Raoul.Rothfeld@glasgow.ac.uk`
`Leif.Azzopardi@strath.ac.uk, and Peter.Triantafillou@glasgow.ac.uk`

[1] University of Glasgow, Glasgow, UK
[2] University of Strathclyde, Glasgow, UK

Abstract. Providing support for users during their search sessions has been hailed as a major challenge in interactive information retrieval (IIR). Providing such support requires considering the context of the search and facilitating the work task at hand. In this paper, we consider the work tasks associated with air traffic analysts, who perform numerous searches using a multifaceted search interface in order to acquire business intelligence regarding particular events and situations. In particular, we develop a novel task completion engine and seamlessly incorporated it within a current air traffic search system to facilitate the comparison of information objects found. In a study with 24 participants, we found that they completed the complex work task faster using the comparison feature, but for simple work tasks, participants were slower. However, participants reported (statistically) significantly higher satisfaction and had (statistically) significantly higher accuracy using the search system equipped with task completion engine. These findings help to steer systems to provide a better support to users in their search process.

1 Introduction

Searching is typically performed in the context of a task (usually a work task) [1, 2], where the user desires to complete the task as efficiently and effectively as possible. While numerous search systems have been proposed to support the search process [3–10], providing effective task support is still a difficult and challenging problem in Interactive Information Retrieval (IIR) [11, 12]. A promising direction to provide task support is the idea of a task completion engine [13], which explicitly goes beyond supporting the search task to facilitating it. This paper is one of the first attempts in this direction. A task completion engine builds on top of a search engine enabling the collection, collation and comparison of information found during the course of a search session. Essentially, the task completion engine aims to augment the user's cognitive capabilities in order to achieve a successful outcome: *reducing task completion times, improving decision making, decreasing the cognitive burden, and crucially reducing errors.*

An important domain where task completion engines can potentially be of great use is within the Air Traffic industry, where finding relevant information efficiently is essential [14]. Flight analysts typically perform complex search tasks in order to find the relevant information (i.e. flight intelligence) to make informed decisions regarding flight performance management. In this context, common work tasks require the analysts to aggregate and compare the different information and data that is available to them. This typically involves posing many queries and examining a number of facets to acquire all the relevant information [14] (and thus is similar to most IIR search/work tasks [1, 2, 15]).

In this paper, we aim to study the effect of task completion engine in the effectiveness and efficiency of users in completing complex search tasks in this domain. To provide a use case for our investigation, we experiment with an air traffic search system [14], where analysts need to interact with information about aircraft, schedules, operators, airports, etc. through textual summaries (i.e. news, weather conditions, traffic conditions, airport notifications, etc.), structured data (i.e. flight times, temperatures, etc.) and visual representations (i.e. charts and graphs, etc.) in a timely manner. To do so, we seemingly incorporated a new feature to a real life air traffic search system to help task completion. Specifically, we proposed a contextualised comparison feature that first enables such systems to store the analysts' search state/results at different points of their search session. Second it allows analysts to compare their current search state/results to the stored one by automatically overlapping (superimposing) them across heterogeneous data visualisation.

This paper has three novel contributions: first we have investigated the effect of task completion engine in the context of a novel and specific domain [16], bringing Information Retrieval techniques to the problem of searching air traffic information [14]. Second we have provided evidence that the introduction of contextualised comparison feature has led to (statistically) significantly higher user satisfaction and accuracy in completing both simple and complex work tasks. Third, we have also found that incorporating task completion engines could introduce both benefits and limitations to search systems depending on the task difficulty faced by the users.

The remainder of the paper is organised as follows. Section 2 describes state-of-the-art works in task completion. Section 3 presents the approach of the paper. Section 4 describes our system. Section 5 and 6 discuss the experimental methodology and results respectively. We then conclude and discuss future work.

2 Related Work

Search engines typically provide only limited support for users across their session(s) and often fail to help users complete satisfactorily more complex information search/work tasks [11, 12]. However, there have been numerous attempts to improve the standard search interface to support searching e.g. [4, 3, 5, 7, 8, 17, 18]. For example, in [8], they augment search sessions by providing a viewable history of the pages that the user has interacted with during the course of a session. The history of pages are shown as thumbnails to provide users with a non-textual cue so that they can quickly re-access previously viewed pages and storing the pages in *WebBooks* [19]. They found that participants used the thumbnail history to view key hub pages, compare information on pages (i.e. hotel prices), and to obtain additional information related to the current page (e.g. to convert a currency). A similar augmentation was developed in [5] called *SearchBar*, which showed the list of pages visited but grouped by query, to enable easier navigation through the result history. In the context of a work task, to organise travel and trips, it was again found that the additional support helped in completing these complex search tasks. Following in this direction was the development of *SearchPad* [17], which was devised to help searchers perform "research missions", i.e. complex search tasks, such as finding a good deal for a HDTV, the value of political parties, or collecting good recipes. *SearchPad* would enable users to take notes about various pages that they encountered through searching, so that they could make sense of the information that they had found, and invariably make a better decision (i.e. on what to buy, who to vote for, what to cook and eat, etc.).

Each of the examples above, highlights the need that people have to use the information that they have previously found in order to perform a work task, and try to augment the search engine/interface to provide cognitive support to help saving, collecting, and re-finding/re-accessing the information. On the other hand, other search interfaces have been devised to help support the exploration of results [3, 4, 9, 10]. For example, Querium [4] provides users with numerous search features such as relevance feedback, query fusion, faceted search, and search histories, and facilitates collaborative search. The idea was to help users share, save, collaborate and revisit their information. *SearchPanel* [3], provides similar functionality through a web browser extension, to support people in their ongoing web information seeking tasks by mapping the space that has been explored. Rather than providing cognitive support in terms of histories and maps, its alternative approach is to help guide the users' querying process by providing facets and faceted search [20, 10, 9]. These interfaces, again, support the users across and through their session as they try to make sense of the information space and achieve a greater awareness of the topic of interest.

These developments have focused on helping users address their work tasks by augmenting the search engine. In [13], Balog sets out a vision for developing task completion engines that during the course of searching extracts out the salient entities and information from the pages, store this information, and facilitate decision making. This requires task modelling, understanding requests, resource representation and selection, and information retrieval, extraction and integration [13]. Key to this process is the information extraction of entities from the pages and the integration of information through semantic analysis with respect to the task at hand. For example, extracting different places to visit when on holidays, the different hotels and deals on offer, the different medicines and treatments available for a particular condition, etc. Thus, the development of such engines requires a significant amount of infrastructure. Here, since we focus on a specific domain, we are able to extract out the salient information based on semi-structured data, and thus can evaluate whether the addition of a contextualised comparison feature facilitates more efficient and effective completion of air traffic analyst work tasks.

3 Approach

As mentioned in Section 1, a task completion (TC) engine usually builds on top of a search engine enabling the collection, collation and comparison of information found during the course of a search session. The aim of such an engine is to augment the user's cognitive capabilities in order to achieve a successful outcome: reducing task completion times, improving decision making, decreasing the cognitive burden, and crucially reducing errors. While the concept of TC engines should by definition benefit users, developing an actual engine with such a functionality is not so easy. This is because this feature needs to be seamlessly merged with already existing functionalities of the search system. This is an important challenge that major search engine companies are facing when they are introducing new features, due to potential damage it can have on the revenue, etc.

With that in mind, we carefully identified an existing limitation in current search systems, i.e. users have to rely too much on their memories to accomplish the work task effectively and efficiently. This situation can become worse when the user has to memorise multiple data points or translate such data points across heterogeneous data representation. In order to tackle this challenge, we introduce the idea of storing search sessions and allowing users to retrieve the stored sessions at any time during the search process. While there exists a wealth of research on retrieving relevant information for a given query, to the best of our knowledge, there is no prior research on providing previous search session states to the user for cross comparison.

We also introduce the idea of highlighting the differences between the data stored and the one for the current search session. This is also a challenging task by itself, since it needs a deep understanding of the problem, various data representation and visualisation techniques that can facilitate users in their complex work task. To investigate our approach, as our use case, we focus on a novel search domain, i.e. an air traffic search system, where users have to perform complex task in a timely manner. In the rest of the section, we discuss how we implemented our approach in an operational air traffic search system.

4 Air Traffic System Task Completion Engine

The standard search engines used by an air traffic control analysis companies [14] are a multifaceted search system, consisting of the standard query input along with facets for selecting airports and airlines, in order to filter data. The results returned contain information objects of various modalities such as the number of flights and flight information including time, day, delays, weather, distances, etc. Such numerical data is extracted out and associated with a particular entity (i.e. a flight, a carrier airline, an airport, etc.) and is used to make various air traffic decisions. While it appears quite different from a standard text based retrieval system, it is more on par with an interface for product search where users can compare prices and technical specifications about products, see ratings, etc. For the purposes of this study, we have seemingly incorporated a task completion engine into an air traffic search system which is representative of those used at a commercial air traffic analysis company. The rest of this section describes components of the system in detail.

Backend Component: During a search session, the backend receives several requests generated through either standard query input or from the interactions with the facets. The backend component then processes the queries, constructs filters, and applies them on the underlying data dimensions and thus, gradually reduces the presented amount of data to the desired subset.

User Interface Component: The search interface (as shown in Figure 1) is composed of a querying interface that contains two drop-down menus, one for airports and the other for airlines (A), a selection reset button (B), and various interactive charts for conducting search queries (C - F). The user is presented with a line chart depicting the number of flight movements over time (C), a scatter plot illustrating all flights according to their time of day and delay in minutes (D), two row charts showing the number of flights per connected airports or weekday (E), and three bar charts showing the number of flights per delay in minutes, time of day, and flight distance in miles (F). Presented data is queried within a search session via mouse interaction on these charts.

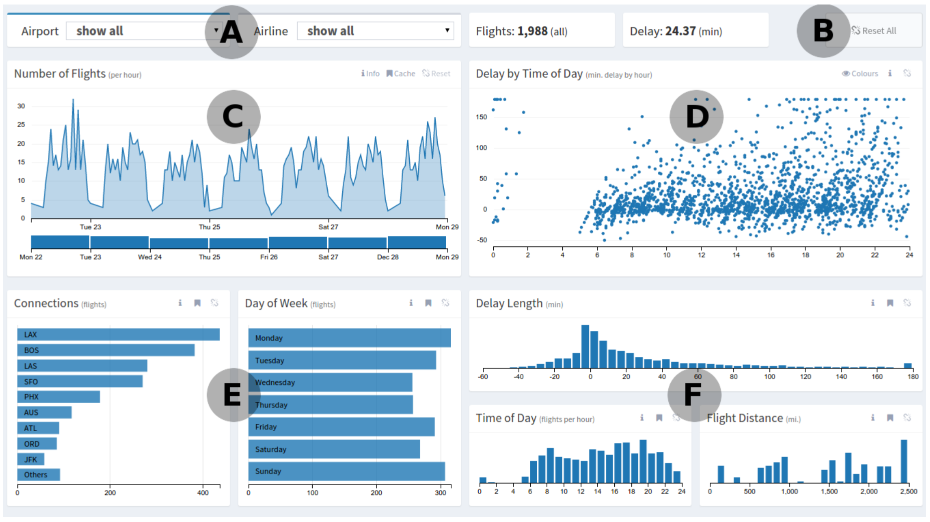

Fig. 1: The interface for air traffic control analysts. (A) Querying Component (B) New Task/Reset (C) Flight Movements Chart (D) Day/Delay Chart (E) Flights per connect airport Charts (F) Flight delay Charts.

User Tracking and Logging component: User actions were monitored and logged by the system, including the number of interactions/clicks and time spent carrying out presented information retrieval tasks. Users were asked to indicate task completion by clicking a designated button.

4.1 Contextualised Comparison Component

The contextualised comparison component can be activated through pressing designated caching buttons above each chart. Upon button press, the current search session and the queried data subset are being saved (see Figure 1, top). Thereon, users can start a new search session and query data according to their interest and re-press the caching button, which will add the novel search session to the comparison component's memory (see Figure 2, bottom). Upon activating the comparison chart, all saved sessions are rendered within a stacked chart overlay (e.g. grouped bar chart or multiple line chart), allowing for contextualised comparisons across search sessions (see Figure 2).

Cross Comparison: Upon caching a search session for later contextualised comparison, the data dimensions and descriptions of all applied filters are saved in a queue. The user's request for comparison renders all cached dimensions within one stacked chart with the filter description of each of the queue's elements as the chart's legend.

In order to add this function seamlessly to the existing factions of the system, we devised a small caching button above any chart allowing contextualised comparison. The system was configured so that the caching buttons, which activated the contextualised comparison component, could easily be hidden from the view.

5 Experimental Set-up

Research Question: The main goal of this study was to investigate the effect of a contextualised comparison feature added to the search interface to facilitate task completion, where we hypothesis that:

- $H1$: providing contextualised comparison will improve the efficacy (in terms of task completion time and number of interactions) and effectiveness (in terms of accuracy of finding the correct answers) of users.
- $H2$: providing contextualised comparison will improve searchers‘ experience (in terms of satisfaction).

Design: This study used a within-subject design, with the independent variables being task difficulty (i.e. from simple lookups / fact finding to more difficult and complex tasks involve numerous queries, data gathering, extracting relevant data/information, and then a comparison) and the availability of the contextualised comparison feature. The dependent variables are the qualitative (gathered through the accompanying questionnaires) and quantitative (gathered through system interaction logging) data. We did not perform any control on the time, number, or type of interactions with the system to simulate a real search scenario as much as possible.

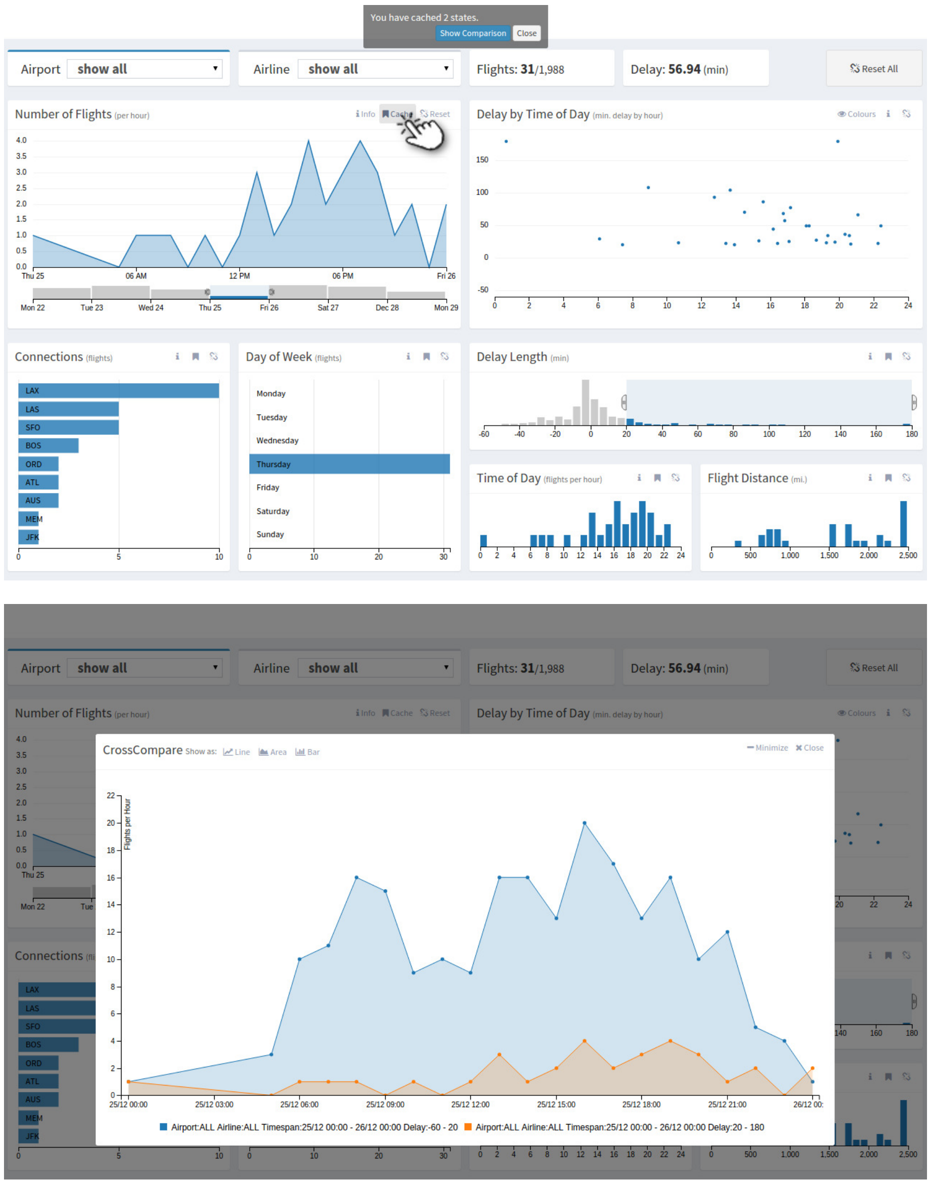
You have cached 2 states.
Show Comparison
Close
Airport show all
Airline show all
Flights: 31/1,988
Delay: 56.94 (min)
Reset All
Number of Flights (per hour)
Info
Cache
Reset
Delay by Time of Day (min. delay by hour)
Colours
Connections (flights)
Day of Week (flights)
Delay Length (min)
Time of Day (flights per hour)
Flight Distance (mi.)
LAX
LAS
SFO
BOS
ORD
ATL
AUS
MEM
JFK
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
CrossCompare Show as: Line Area Bar
Minimize
Close
Flights per Hour
Airport:ALL Airline:ALL Timespan:25/12 00:00 - 26/12 00:00 Delay:-60 - 20
Airport:ALL Airline:ALL Timespan:25/12 00:00 - 26/12 00:00 Delay:20 - 180

Fig. 2: Top: The interface after the use has saved one query in the session. Below: A comparison between the cached queries and the current query.

Data and Materials: The air traffic search system is built on a reduced set of flight entries obtained from the American Statistical Association[3], which comprises heterogeneous flight information (such as origin and destination, date and time, airline code, or delays) for flights within the Unites States from 1987

[3] `http://www.amstat.org/publications/jse/jse_data_archive.htm`

to 2008. The various types of data are being made available to the user via the different charts of the user interface.

Tasks: Our commercial partner supplied a number of typical work tasks performed by their analysts. Using a similar approach to Brennan et al. [21], we selected ten work tasks, five of which we considered simple, and five of which we considered complex. The difficulty of these tasks was determined during a pilot study and measured by the number of interactions required to complete the task (i.e. apply query filter, view information object, note information/data, etc.). The complex tasks require participants to perform multiple queries and compare/contrast the information gathered from various filter states for each of these queries with each other to draw a final conclusion. Simple tasks can be answered via issuing a single query. Examples of the two types of tasks are:

Simple *How many flights were operated per weekday?* This task required the participant to apply no filters.

Simple *How many American (AA) airline flights were operated during the busiest hour at Chicago (ORD) airport, which had a delay of 0-60 and 60-120 minutes?* This task required the participant to apply one airline and one airport filter, while switching between two delay filter states.

Complex *For each day of a week, which airport has the most flights per weekday?* This task required the participant to cycle through all airports and contrast the weekday values.

Complex *On Thursday the 25th , what is the difference in number of flights with a delay of less than 20 minutes compared to the number of flights with a delay of 20 or more minutes; at 8, 12, 16, and 20 o'clock?* This task required the participant to apply one time filter and compare the two different delay filter states at four points in time.

To counteract the order and fatigue effects we counter-balanced the task distribution using a Graeco-Latin Square design.

Procedure: The ethics approval was obtained from the University of Glasgow. The formal meeting with the participants took place in an office setting. At the beginning of the session the participants were given an information sheet which explained the conditions of the experiment. The participants were notified that they have the right to withdraw from the experiment at any point during the study, without affecting their legal rights or benefits then asked to sign a Consent Form. Then, they were given an Entry Questionnaire to fill in.

The session proceeded with a brief tutorial on the use of the search interface with a short training task. After completion of the training task, each participant had to complete six search tasks (see Section Tasks), one for each level of task difficulties where the comparison feature is available or not (see Section 5). To negate the order and fatigue effects we counter-balanced the task distribution using a Graeco-Latin Square design.

The subjects were given 10 minutes to complete their task, during which they were left unattended to work. At the end of each task, the subjects were asked to complete a post-task questionnaire. Questions in the post-task questionnaire were randomised to avoid the effect of fatigue. Between each task, a cooling-off period was applied to avoid the carry-over effect. Finally, an exit questionnaire was administered at the end of the session.

Each study took approximately 120 minutes to complete; this is from the time they accepted the conditions until they finished answering the exit questionnaire. Users could only participate once in the study. The participants were all volunteers and did not received any compensation. The results of these studies are presented in Section 6.

Participants: 24 people were recruited to undertake the study, of which 8 were female and 16 were male. All participants were between 18 and 64 years old, with most between 18-24 (62.59%) and then 25-34 (25.00%). Most participants had at least bachelor degree (83.33%) at the time of the experiments. The majority of participants had knowledge about Computing and Information Technology, in particular search systems.

Baseline vs. Enriched System: For experimental purposes, we used two versions of the system, one without contextualised comparison component (i.e. Baseline) and one with (i.e. Enriched). The changes in the user interface (UI) between these two systems are minimal to avoid introducing any confound effect. In particular, for the Baseline system, the UI consists of all components, as shown in Figure 1, without caching buttons above any chart – rendering the comparison functionality inaccessible. Whereas the UI of the Enriched system facilitates the use of the contextualised comparison component via caching buttons above each chart and, consequently, the comparison chart overlay.

Apparatus: For our experiment we used one desktop computer, equipped with a monitor, keyboard and mouse. The computer provided access to a custom-made air traffic search system which allowed the participants to perform their search tasks. The system was designed such that it logged participants' desktop actions, such as starting, finishing and elapsed times for interactions, mouse clicks using a common system time.

Questionnaires: At the beginning of the experiment, the participants completed an *entry questionnaire*, which gathered background and demographic information, and inquired about previous experience with online search systems and searching air traffic control data. At the end of each task, the participants completed a *post-task questionnaire*, where they were asked about their satisfaction with the system. Finally, an *exit questionnaire* was introduced at the end of the study gathering information about their general comments about the experiment.

6 Results

To compare the differences between the two systems we performed a paired t-test between the various measures taken for each system to check whether the Enriched system (i.e equipped with contextualised comparison component) was significantly different to the Baseline system. We use (*) and (**) to denote the level of significance where the confidence level is ($p < 0.05$) and ($p < 0.01$), respectively.

Log Analysis: Table 1 reports the mean (and standard deviation) of the time taken to complete the simple and complex tasks, along with the number of interactions (i.e. queries, clicks, facets, etc) as well as the accuracy[4] at performing the tasks.

Table 1: Mean completion times, no. of interactions and accuracy per task. The value in parenthesis is the standard deviation. (*) and (**) denotes difference with the confidence levels ($p < 0.05$) and ($p < 0.01$) respectively.

Task	Task completion time		Interactions		Accuracy	
	Baseline	Enriched	Baseline	Enriched	Baseline	Enriched
Simple	**133.54** (80.34)	159.25 (96.55)	**15.92** (6.55)	26.87 (17.56)	83.33% (28.86)	**100%**** (0.0)
Complex	416.39 (212.30)	**156.38**** (78.35)	88.89 (38.12)	**23.55**** (14.13)	77.77% (25.45)	**88.88%*** (9.62)

The results indicate that for simple tasks (i.e. lookup based task), participants on the Baseline system completed the task with fewer interactions (15.9 vs 26.9) and did so in less time (133.5 vs 159.3 seconds). This could be due to participants' expertise with the Baseline system, although both results were not statistically different.

However, for the complex tasks, our results suggest that participants on the Enriched system performed significantly fewer interactions (23.5 vs 88.9) and completed their tasks in significantly less time (156.4 vs 416.4 seconds). In this case, both results were statistically different suggesting that for the more complex tasks that required participants to memorise several data points and cross compare them, the Enriched system provides a clear advantage. It appears that using the contextualised comparison component resulted in a slower performance in simple tasks but a quicker performance in complex tasks (addressing RQ1).

Interestingly, the participant's accuracy in performing their tasks significantly improved for both simple and complex tasks when the contextualised comparison component was used. These results were statistically different suggesting that participants made consistently fewer errors with the Enriched system. Our findings show that our task completion engine improved participants' effectiveness in performing their complex tasks which in such a domain could be extremely important (addressing RQ1). We now turn our attention to the questionnaire analysis to see if it reveals any further insights.

Questionnaire Analysis: 23 out of 24 (95.8%) participants reported that they preferred using the Enriched system. Further all participants found it to be somewhat or very helpful. In addition, the majority of participants felt it was somewhat or very intuitive, except two participants (8.3%).

In terms of satisfaction, we asked participants to rate how easy it was to complete tasks with each system and how satisfied they were with the amount of time it took to complete tasks with each system. Table 2 shows the results for satisfaction, where the Enriched system was rated significantly higher on both

[4] The ratio of the number of correct answers to the total number of answers given.

counts (addressing RQ2). These suggests that even though participants took a little bit longer on average for simpler tasks they did not detract from their rating with respect to how satisfied with the time to complete tasks.

Table 2: Mean user satisfaction (SAT) per task on ease of completion and required amount of time, 1 (strongly disagree) to 5 (strongly agree). The value in parenthesis is the standard deviation. (**) denotes difference with the confidence levels ($p < 0.01$).

SAT with ease of completion		SAT with amount of time	
Baseline	Enriched	Baseline	Enriched
2.89 (1.22)	**4.66** (0.716)**	2.77 (1.14)	**4.72** (0.55)**

Comments from participants also confirmed this as they mentioned that it was "*easier*" and "*faster*" to complete tasks using Enriched system, while others mentioned that some of the complex tasks were "*laborious*" and "*infuriating*" to complete without the contextualised comparison component. "*[The] ability to store and then compare information significantly aided its interpretation*", stated one participant with others agreeing that the contextualised comparison component was "*ideal for complex querying*" and for filtering out "*the factors that matter to you*". However, multiple participants stated that the comparison feature did not benefit the completion of simple tasks. Others participants mentioned that even for simpler tasks the contextualised comparison component was useful as it enabled them to double check their results. This last comment was kind of unexpected, but suggests that the comparative component is useful to ensure accuracy.

7 Discussion and Conclusion

This paper investigated the effects of task completion engine on the efficiency, effectiveness and satisfaction of participants in completing complex work tasks. As a use case scenario, we considered the work tasks associated with air traffic analysts, who perform numerous searches in order to acquire business intelligence regarding particular events and situations. To support their work tasks, we seemingly incorporated such an engine, a contextualised comparison feature, into an air traffic search system.

Our findings reveal that participants on the search system equipped with contextualised comparison feature (Enriched system) completed complex tasks much more efficiently. However, on simpler tasks our participants took longer time to do so. This appeared to indicate that the Enriched system hindered their efficiency, but participants reported that they checked their answers, which took more time but ensured greater accuracy. Crucially participants had (statistically) significantly higher satisfaction and accuracy using the Enriched system. These findings show that introducing additional features such as contextualised comparison is generally positive, but it may increase the time to complete simpler tasks. This suggests that as we propose novel task completion engines, we need to be careful to determine when they help and when they hinder the user.

One of the main limitations of our study is that it is domain-specific and the use case of an air traffic search system is a rather industry-specific application. However, the notion of contextualised comparisons and search session caching can be applied to a multitude of information retrieval scenarios. Thus, it is expected to improve users' search sessions experiences in a wide range of information seeking tasks and lessen the user's cognitive load. However, this may come at the cost of reducing the efficiency at simpler tasks. Nonetheless, we have provided strong empirical evidence that the concept of contextualised comparison improves the search experience, efficacy and effectiveness lending weight to the progression from search engines towards task completion engines. Further work will be directed towards developing similar contextualised comparison component for other domains and tasks.

Acknowledgement: This work was supported by the Economic and Social Research Council [grant number ES/L011921/1].

References

1. Vakkari, P.: Task-based Information Searching. Ann. Rev. Info. Sci. Tech. **37**(1) (2003) 413–464
2. Toms, E.G., Villa, R., McCay-Peet, L.: How is a search system used in work task completion? J. Information Science **39**(1) (2013) 15–25
3. Qvarfordt, P., Tretter, S., Golovchinsky, G., Dunnigan, T.: SearchPanel: Framing Complex Search Needs. In: SIGIR '14, ACM (2014) 495–504
4. Diriye, A., Golovchinsky, G.: Querium: A Session-based Collaborative Search System. In: ECIR'12. (2012) 583–584
5. Morris, D., Morris, M.R., Venolia, G.: SearchBar: A Search-Centric Web History for Task Resumption and Information Re-finding. In: CHI '08, ACM (2008) 1207–1216
6. Sebrechts, M.M., Cugini, J.V., Laskowski, S.J., Vasilakis, J., Miller, M.S.: Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces. In: SIGIR '99, ACM (1999) 3–10
7. Jones, W., Bruce, H., Dumais, S.: Keeping Found Things Found on the Web. In: CIKM '01, ACM (2001) 119–126
8. Jhaveri, N., Räihä, K.J.: The Advantages of a Cross-session Web Workspace. In: CHI EA '05, ACM (2005) 1949–1952
9. Villa, R., Gildea, N., Jose, J.M.: FacetBrowser: A User Interface for Complex Search Tasks. In: MM '08. MM '08, New York, NY, USA, ACM (2008) 489–498
10. Kashyap, A., Hristidis, V., Petropoulos, M.: FACeTOR: Cost-Driven Exploration of Faceted Query Results. In: CIKM '10, ACM (2010) 719–728
11. Belkin, N.J.: Some(What) Grand Challenges for Information Retrieval. SIGIR Forum **42**(1) (June 2008) 47–54
12. Gäde, M., Hall, M.M., Huurdeman, H., Kamps, J., Koolen, M., Skove, M., Toms, E., Walsh, D.: Report on the First Workshop on Supporting Complex Search Tasks. SIGIR Forum **49**(1) (June 2015) 50–56
13. Balog, K.: Task-completion Engines: A Vision with a Plan. In: Proceedings of the First International Workshop on Supporting Complex Search Tasks. Volume 1338. (2015)
14. Billings, C.: Aviation Automation: The Search for a Human-Centered Approach. Human factors in transportation. Lawrence Erlbaum Associates Publishers (1996)
15. Ruthven, I.: Interactive Information Retrieval. Annual Rev. Info. Sci & Technol. **42**(1) (January 2008) 43–91
16. Salampasis, M., Fuhr, N., Hanbury, A., Lupu, M., Larsen, B., Strindberg, H.: Integrating IR Technologies for Professional Search. In: ECIR '13. (2013) 882–885
17. Donato, D., Bonchi, F., Chi, T., Maarek, Y.: Do You Want to Take Notes?: Identifying Research Missions in Yahoo! Search Pad. In: WWW '10, ACM (2010) 321–330
18. Dziadosz, S., Chandrasekar, R.: Do Thumbnail Previews Help Users Make Better Relevance Decisions About Web Search Results? In: SIGIR '02, ACM (2002) 365–366
19. Card, S.K., Robertson, P.G.G., York, W.: The WebBook and the Web Forager: An Information Workspace for the World-Wide Web. In: CHI '96, ACM (1996) 111–ff.
20. Tunkelang, D.: Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers (2009)
21. Brennan, K., Kelly, D., Arguello, J.: The effect of cognitive abilities on information search for tasks of varying levels of complexity. In: IIiX '14, ACM (2014) 165–174