Seeding Simulated Queries with User-study Data for Personal Search Evaluation

David Elsweiler[1], David E. Losada[2], José Carlos Toucedo[2], Ronald T. Fernández[2]
[1]Department of Computer Science (8 AI), University of Erlangen, Germany
david@elsweiler.co.uk
[2]Departmento de Electrónica y Computación, Universidad de Santiago de Compostela, Spain
{ david.losada, josecarlos.toucedo, ronald.teijeira }@usc.es

ABSTRACT

In this paper we perform a lab-based user study (n=21) of email re-finding behaviour, examining how the characteristics of submitted queries change in different situations. A number of logistic regression models are developed on the query data to explore the relationship between user- and contextual- variables and query characteristics including length, field submitted to and use of named entities. We reveal several interesting trends and use the findings to seed a simulated evaluation of various retrieval models. Not only is this an enhancement of existing evaluation methods for Personal Search, but the results show that different models are more effective in different situations, which has implications both for the design of email search tools and for the way algorithms for Personal Search are evaluated.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]:

General Terms

Measurement,Experimentation, Human Factors

Keywords

Personal Search, Email Re-finding, User Study, Evaluation

1. INTRODUCTION

It is well documented in the literature that people regularly need to re-access and re-use information that they have created or accessed in the past [11, 27, 29] and that existing desktop management and search tools are inadequate to support this activity effectively, resulting in huge frustration and waste of resources [1, 4, 5, 28]. Personal Information Access (PIA) as a research area focuses on providing search solutions to help people re-find information they have seen or accessed previously or information relating to themselves [16]. The need for better tools to support PIA is clear. However, there are two main research issues that must be addressed to facilitate progress in this area:

First, little is known about the behaviour of users in response to information re-access needs, i.e. the behaviour that search tools should support. An understanding of what people need and how they act in order to achieve those aims is essential to know how to best support user behaviour, either through algorithmic or interface support.

Second, new tools and algorithms are difficult to evaluate scientifically given a lack of open and accepted test collections and evaluation frameworks for Personal Search. Although progress in this area has been made [3, 20, 21], the current state of the art approach suffers from a number of limitations and is not widely used as a result.

In this paper we contribute to resolving these issues. First, we perform a user study to learn about the characteristics of user querying behaviour when re-finding email messages and the factors which can influence the types of queries people submit. We examine the relationship between user and contextual variables and query characteristics including the length of the query, the field submitted on and the use of named entities. Second, we utilize the findings of this study to improve the query simulation process as applied in the literature [20, 21]. We analyse the performance of several retrieval algorithms based on simulated query profiles generated from the user study data. Taking this approach allows us to investigate and understand the relationship between user behaviour and algorithmic support for re-finding.

The remainder of the paper is structured as follows: Section 3 presents the user study, detailing the study design and data analyses; Section 4 describes the seeded evaluation process, explaining the choice of collections, the query generation method, and the retrieval models tested; Section 5 presents the experimental results; Section 6 discusses the scope and limitations of our work. Finally, Section 7 outlines our main conclusions and plans to extend the work in the future. First, to motivate the work and provide a platform to discuss our findings, we summarize appropriate related work.

2. RELATED WORK

2.1 An Overview of Re-finding behaviour

Previous work has shown people to re-access and re-find information regularly. For example, 60-80% of web page visits are re-accesses [23] and roughly 40% of web searches are performed with the aim of re-finding something seen before [27]. Desktop search tool logs show that on average

SIGIR'11, July 24–28, 2011, Beijing, China.

users submit over 4 queries per day and email is the type of media people re-find most often [11, 10].

Re-finding queries tend to be much shorter than typical web queries. Dumais and her colleagues [11] and Cutrell and his colleagues [10] both report average queries lengths of 1.6 terms compared to the well documented 2.3 for web search. Tyler and Teevan [29] found that web re-finding queries had on average 12.1 characters compared to 18.9 for queries to find new results. Desktop search logs show that re-finding queries rarely contain advanced query operators with as little as 7.5% of the queries containing features, such as boolean operators, phrases, or field restrictions specified [11]. Named entities play an important role in re-finding with queries often containing references to people. A quarter of queries in [11], for example, contained people's names.

Context is another important aspect of re-finding. What people tend to remember [13] and how difficult they perceive re-finding tasks to be [15] both vary in different situations. Given these relationships, it is possible that different behaviour will be exhibited in these situations as has been shown previously for web search engine behaviour. In web search, when users are faced with a difficult task, they start to formulate more diverse queries, use advanced operators more, and spend longer examining result pages [2]. If similar behavioural changes occur when performing difficult re-finding tasks then it would important consequences for the way behaviour is simulated in automated evaluation approaches for Personal Search. Here we seek to identify important situational variations to inform the choice of tasks in simulated evaluations.

2.2 Evaluating Personal Search

The difficulties in evaluating Personal Search behaviour are well-documented [17, 19]. In addition to major privacy and participation issues, there is a lack of shared resources, such as test collections and tasks. The main approaches to date have been naturalistic investigations with specific tools [11, 10, 9], case studies for particular scenarios [19] and lab-based user studies [7, 25, 17]. While all of these approaches offer advantages, none are suited to the controlled, repeatable evaluation of search algorithms.

Methods for automated evaluation have also been proposed. For example, Chernov and colleagues [8] suggested that researchers volunteer their own personal data to create a shared test collection for research purposes. Kim and Croft use pseudo-desktop collections that share the properties of personal collections to avoid privacy issues[20]. Three separate collections were created from the TREC Enterprise Track's W3C dataset, by taking the email messages for three prominent people in the collection and augmenting them by gathering related documents of various types from the Web. A further collection was created by using public university documents relating to individuals. These datasets have since been publicly released.

Using their collections, Kim and Croft create known-item retrieval tasks based on simulated queries [3]. Simulated queries are a potentially powerful method of scientific evaluation for Personal Search. However, there are problems with current implementations, which are over-simplified and make assumptions about user behaviour that are not necessarily true. For example, query terms are typically drawn independently from the document and either do not make use of field information [3] or assume that all fields are equally likely to be queried on [20]. Further, current implementations do not incorporate what we already know about user behaviour (e.g. that people often make use of named entities in queries).

It seems likely to us that the kinds of queries submitted will change in different scenarios. Re-finding behaviour is guided by user recollections [6] and people remember different things in different situations, with this being heavily influenced by contextual factors [13]. We hypothesize that the types of queries submitted will change in different situations. If this is the case, not only should this be incorporated in simulations, but it may mean that the type of algorithmic support required will also vary situationally, with obvious implications for search tools.

To test this hypothesis we performed a controlled user study to examine re-finding behaviour for email messages. We analyse some of the data collected to learn about the characteristics of the queries submitted and how these can change in different situations.

3. A STUDY OF EMAIL QUERY BEHAVIOUR

3.1 Study Design

We decided to focus on email search tasks because looking for different types of documents may lead to different behaviour and therefore require a larger-scale study to investigate properly. As email is the type of object re-found most often and there are appropriate collections available, we felt this would be a good starting point for research of this kind.

Our study population included 21 participants from a well-known British university, consisting of a mix of academic and research staff, undergraduate computer science students and a post-graduate class with a variety of undergraduate academic backgrounds, including former business, geography, modern-languages and philosophy students. The participants had been using their collections for varying time periods, with the post-grads having relatively new collections (the average age of collection was less than 3 months) and the academic staff comparatively older collections (avg. age ~3years). Reflecting this, some of the collections contained few messages (min = 95) and others several thousand (max =8954, median = 5132). The participants also reported using email for different purposes. While the students tended to use email mainly for class announcements and collaborative working, the academics used email for a wide range of purposes, including task and contact management, data storage, version control, collaborative authoring, as well as simple communication.

We went to great lengths to establish realistic re-finding tasks for participants that could be performed on their own personal collections without invading individual privacy. This was achieved following the methodology proposed by [17] and involved performing a number of preliminary studies with the participants and their peers, including interviews, collection tours and diary studies of the re-finding tasks people in these groups perform. This work allowed us to establish a pool of experimental tasks suitable for each groups of users. These pools reflected the contents of their collections and simulating the kinds of re-finding tasks they may perform in a naturalistic setting. The task pools contained

example tasks of each of the three types identified in [17]: Lookup tasks involved finding specific pieces of information, such as passwords or phone numbers from an email; Item tasks involved finding complete emails perhaps to print out or forward to someone else; and multi-item tasks involved re-finding multiple email messages and sometimes processing to content of those mails to complete the task[1].

Each participant was allocated 9 tasks from these pools to complete on 3 systems (3 tasks per system - 1 of each type), 2 search-based interfaces and a third interface where the participants could only browse through their folders to find the information required to complete tasks. Here, we only study the queries submitted to the search systems because we were interested in understanding how the characteristics of submitted queries changed in different situations. Both search interfaces provided an interface widget to select the field that the query would be submitted to, but it is important to acknowledge that the search systems were not the same, differing in the way queries were submitted and in the way that results were presented. The main difference was in the way results were presented. The first system presented results as a standard list, while in the second system results were clustered graphically by date received. Full details of the systems can be found in [12]. We account for differences in the systems in our data analyses below.

The responses from pre- and post-task questionnaires combined with the demographic data and collection statistics for the experimental population provided a rich basis to investigate the variables influencing the querying behaviour[2]. Further details of the experimental design and user population can be found in our previous publications [13, 14], which analysed different aspects of participant behaviour.

3.2 Data Analyses

In total 347 queries were submitted. The mean length of the queries was 1.48 words (max = 7). A good mixture of fields were queried on. The most commonly queried on field was the sender field (39.48% of queries contained at least one clause on sender field), the least common field was "to or cc", which only featured in 7.5% of queries. Only 13.3% of queries were submitted against all fields. Named Entities (NEs) were heavily used with 60.5% of queries containing a reference to the name of a person, place, event or thing. Peoples names were most common, featuring in 40% of all queries and 24.21% of all queries contained a NE other than a person's name.

To understand the influence various contextual factors had on the characteristics of submitted queries we developed a number of logistic regression models. Logistic regression is a useful way of describing the relationship between one or more independent variables (e.g., the number of emails in a collection or the user filing strategy) and a binary response variable that has only two possible values expressed as a probability, such as ("contains a NE" or "does not contain a NE"). We were interested in several query characteristics including, the field the query was applied to (e.g. "contains a clause for Sender field"), types of named entities contained within the query and query length (i.e, whether the query is longer or shorter than the mean value). There were other factors of interest in the logs. For example, spelling mistakes were obviously present in some queries and expert query syntax was sometimes used to exploit email etiquette, e.g. "Fwd" or "Re" in the subject line. However, examples of these kinds were too rare to be considered in the analyses.

In total 7 models were generated. All available factors (24 in total) collated from the user study were analysed initially using a stepwise procedure in order to isolate any significant relationships. The stepwise procedure automatically enters and removes factors at each step assessing the overall goodness of fit of the linear regression model ([22] provide an overview on generalized linear models and the stepwise procedure). As an example, Table 1 presents the regression model associated with the length of the query in words. The remaining models can be found in the Appendix. These other models are associated with the following query characteristics: hasSender (whether or not the query contains a clause on the Sender field), hasSubj (whether or not the query contains a clause on the Subject field), hasBody (whether or not the query contains a clause on the Body field), NE (whether or not the query contains a NE), Person_NE (whether or not the query contains a NE person), and Other_NE (whether or not the query contains a NE -other than person-). Examining these tables shows a number of contextual variables, some of which are highlighted in bold. While all of these variables contribute to the model's predictive power, the significant factors (marked in bold), are those that exert the biggest influence. For instance, the query length model (Table 1) shows that, in our study, having an old collection significantly influenced the length of queries submitted, while other variables, such performing the task frequently did not.

The generated models indicate that several variables had an influence on the users' querying behaviour. Here we focus on variables that featured significantly in several of the models developed: Collection age (whether the collection was new (< 1 year old), medium (up to 2 years old) or old (> 2 years old); Task temperature [26](if the sought-after information was hot (had been accessed in the last week), warm (accessed in the last month), cold (had not been accessed for over 1 month) or range, where multiple emails needed to be re-found and no temperature category fitted; Task difficulty (high vs medium vs low); User experience (high vs low); and User filing strategy (filers vs no filers vs spring-cleaners [30])[3].

Table 2 summarizes the important findings for all of the models generated, showing the contextual variables that were significant factors in many models and how these factors influenced the characteristics of submitted queries.

Examining these variables reveals several interesting trends:

- Participants with older collections tended to submit longer queries and were more likely to query on the subject field. The age of the collection also seemed to influence the way NEs were used in the queries. For example, queries submitted against older collections were much more likely to contain a NE than those submitted against newer collections.

- Experienced participants were more likely to query on the subject field than less-experienced participants, but were less likely to query on the sender field.

[1]The exact experimental tasks can be found in [12].

[2]The full questionnaires can be found in [12].

[3]Task difficulty and user experience were derived from the participants' responses in the questionnaire.

Variable of Interest		length	hasSender	hasSubj	hasBody	NE	Person_NE	Other_NE
General profile		1.48	0.39	0.31	0.24	0.61	0.39	0.24
Collection Age	new	1.37	-	0.33	-	0.55	-	-
	medium	1.47	-	0.23	-	0.58	-	-
	old	1.64	-	0.42	-	0.71	-	-
Task Temperature	Hot	1.68	-	0.43	0.16	-	0.27	0.4
	Warm	1.29	-	0.34	0.16	-	0.46	0.28
	Cold	1.47	-	0.28	0.39	-	0.43	0.15
User Experience	high	-	0.39	0.34	-	-	-	-
	low	-	0.4	0.26	-	-	-	-
Task Difficulty	difficult	1.47	0.28	-	-	-	-	-
	medium	1.53	0.48	-	-	-	-	-
	easy	1.46	0.46	-	-	-	-	-
Filing Strategy	Filers	-	-	-	0.29	-	-	0.08
	no filers	-	-	-	0.31	-	-	0.3
	spring cleaners	-	-	-	0.09	-	-	0.21

Table 2: A summary of the main findings of modelling process. The main variables of interest are shown with their influence on various query characteristics. The figures represent the mean value of the query characteristic (- means the variable did not significantly influence the characteristic so the general profile value is used when generating queries).

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	-2.72579	1.77215	-1.538	0.12497
Collection age:new	0.08563	0.12551	0.682	0.49556
Collection age:old	**0.25779**	**0.12418**	**2.076**	**<0.05**
Avg.#emails/day	**0.03421**	**0.01307**	**2.617**	**<0.01**
Task-type:item	-0.06530	0.13495	-0.484	0.62880
Task-type:multi	0.24317	0.16124	1.508	0.13248
Temperature:hot	0.11253	0.16204	0.694	0.48787
Temperature:range	-0.25784	0.20498	-1.258	0.20932
Temperature:warm	-0.25499	0.14462	-1.763	0.07879
Task Freq:infreq.	0.05257	0.12724	0.413	0.67977
Task Freq:freq.	-0.26170	0.14921	-1.754	0.08038
Difficulty:hard	**3.81902**	**1.66846**	**2.289**	**<0.05**
Difficulty:easy	**3.70450**	**1.66782**	**2.221**	**<0.05**
Difficulty:medium	**3.78639**	**1.65860**	**2.283**	**<0.05**
Sender	0.18179	0.12036	1.510	0.13189
Reason	0.20308	0.13210	1.537	0.12518

Table 1: Regression Model for Query Length

- When the participants were looking for older information they were much more likely to query on the body field. When they were looking for newer information the queries submitted were less likely to contain a reference to a person, but more likely to include other NEs such as organisations, books, groups or programming languages. This situation was reversed for older information with queries submitted in these situations more likely to contain references to people.

- There was a correlation between the task difficulty rating and the probability of the query including a reference to a person with easier tasks more likely to have such a reference. Difficult tasks were much less likely to be on the sender field. There was the unusual finding that tasks rated as being of medium difficulty tended to have some different characteristics to those rated as easy or difficult. For example, tasks of medium difficulty tended to be longer (mean =1.53 words) than easy tasks (mean =1.46) and difficult tasks (mean =1.47).

- The filing strategy that the participant applied also seemed to influence their querying behaviour. Participants with more than 1 folder were much less likely to query on the body of an email than those who kept all of their messages in the inbox. Further, of those participant who did use folders, those who tended to tidy up their folders occasionally (spring-cleaners), were less likely to query on the body of the email than those who filed emails regularly.

The main conclusions from these analyses are that: 1) querying behaviour changes in different situations, and 2) clear relationships exist between certain variables and the query submitted in terms of length, field to which the terms were submitted and the use of NEs in the query.

In the next part of the paper we use these results as the basis of what we argue is an improved query simulation process to better understand how different retrieval algorithms perform in the various situations shown to be important in these analyses.

4. QUERY SIMULATION FOR PERSONAL SEARCH

4.1 Email Collections

The simulated evaluation process requires the selection of an appropriate collection. Here we use four such collections. The first collection is the set of emails contained in the CS collection described in [21]. This is a collection of 806 emails from a computer science department's mailing list. Three further collections were generated from TREC Enterprise track dataset by following the methodology used in [20]. This involved filtering the W3C mailing list collection where the name of each person was tagged, enabling us to identify prominent individuals. We chose three such individuals and isolated the messages sent and received by them to create three unique collections (W3C_U1: 3943 emails, W3C_U2: 3152 emails, and W3C_U3: 1892 emails). These collections share many of the properties of personal collections and thus seem a reasonable way to simulate personal collections without compromising privacy.

4.2 Query simulation

Strategies for building simulated queries have been proposed for known-item web page search [3] and for desktop search [20]. Essentially, they are based on randomly selecting a document (known-item) from the collection and algorithmically selecting query terms from the target document. This leads to the automatic generation of simulated queries and relevance judgments. These methods have been shown

to be very effective and have been evaluated successfully under different dimensions (i.e predictive and replicative validity [3, 20]). This simulation approach is appropriate in our case because we work with email re-finding queries, which are typically known-item queries [17] and are a sub-problem of desktop search [20].

We have adapted and improved these techniques in a number of ways. Current simulations are over-simplified and make assumptions about user behaviour that are not necessarily true. For example, query terms are typically drawn independently from the document and either do not make use of field information [3] or assume that all fields are equally likely to be queried on [20]. Our analysis in Section 3.2 demonstrates that the presence of fields in real queries is not uniform and the resulting statistical models allow the generation of queries incorporating appropriate statistics for the presence of terms from different fields. A further problem with current implementations is that they do not incorporate other important aspects of user behaviour (e.g. that people often make use of named entities in queries). Neither [3] nor [20] consider any kind of named entity information to guide the query production process. We claim that this is problematic because the literature shows that queries for known-item tasks contain a high number of named entities [11, 10], a fact that is further evidenced in our data. Therefore, biasing the query generation towards named entities is likely to produce more realistic queries.

4.2.1 Simulating email re-finding queries

The data analyses in Section 3 revealed that a number of variables influenced the characteristics of submitted queries [See Table 2]. For instance, different collection ages led to different patterns in variables, such as query length or presence of named entities. In contrast, other query characteristics were unaffected by this contextual variable. The data show, for instance, that different collection ages did not influence the probability that the query contained a clause submitted to the Sender or Body fields, nor whether the query was more likely to contain a person or another kind of NE. The simulation process has to account for this.

Another important point to note is that we aim to compare an assorted set of retrieval models. This means our simulation should generate flat queries, consistent with the approaches in the literature [3, 20]. The fact that the trends in Section 3 come from analyses of query logs where the queries are often structured is not a problem because these trends can be used to infer the fields at which particular terms were targeted and, therefore, help us to produce simulated flat queries with query terms in a way that reflects real-life behaviour.

In order to replicate queries associated to the situations described in Table 2 our simulation proceeds as follows:

a) Obtain a *general model* from the complete query log that reflects the general statistics found empirically for all the target variables (length, hasSender, hasSubj, hasBody, NE, Person_NE, Other_NE). For instance, if we randomly draw a one-item sample from this general model we could obtain $(3, 1, 0, 1, 1, 1, 0)$, which essentially says that we need to produce a 3-term query the terms for which should come from multiple fields (Sender+Body fields). The query should also contain a named entity that refers to a person.

b) For each situation (i.e. each potential value for each of the variables of interest), obtain a *situation-dependent model* that biases the query generation process towards the pattern determined by the situation. For instance, the value of the collection age variable strongly influences the following query variables: length, hasSubj, and NE. Therefore, for the three possible values of the collection age variable (old, medium, new) we obtain three situation-dependent models that give us proper statistics for the three query variables. Again, this is computed from the distribution obtained empirically from our user study query logs.

Next, we produce artificial queries for each situation of interest by repeating the following process:

1. randomly draw one item from the situation-dependent model
2. obtain a random item from the general model whose *situation-influenced variables* fit the pattern extracted in 1. This gives us the statistics needed for the query to be generated (*target query*).
3. initialise an empty query.
4. randomly select a document d_i from the collection as a known-item.
5. if the target query demands a NE then randomly extract a NE from any of the target fields of d_i (fields set to 1 in the target query statistics)[4]. If there are no NEs in the target fields or the number of terms in all possible NEs exceeds the target query length then go back to 2.
6. complete the query (up to the target length) with (non NE) terms extracted randomly (popularity selection as described in [3]) from the target fields. This completion should ensure that all target fields set to 1 are satisfied (i.e. they all contribute at least one term to the query). If this is not possible then go back to 2.
7. record the query-doc pair in the relevance judgments file and store the query in the query file for proper evaluation.

For each situation (i.e for each possible value of the variables of interest [See Table 2]) we generated and evaluated 10 sets of 100 simulated queries and we report the average performance obtained over the 10 query sets.

4.3 Retrieval models

In our experiments we evaluated an assorted set of retrieval algorithms: a) the well-known **bm25**, which ignores any document structure and represents the email as a flat bag of words; b) Language Models (LMs) based on Query likelihood [32]. Here, we considered both Jelinek-Mercer and Dirichlet smoothing and tested the following alternatives: **lm email**, where the document LM is constructed from all document fields (considering the doc as plain text), and **lm body**, **lm from**, **lm to**, and **lm subject** where the models are constructed from a single field of the document (and the remaining fields are ignored); c) a well-known mixture-based model (**lmmix**), as described in [24]. This defines the

[4]The collections were preprocessed using the Stanford Name Entity Recognizer, which automatically detects NEs and distinguishes between particular types of NEs (including person, location and organization) and is available at http://nlp.stanford.edu/software/CRF-NER.shtml

document's LM as a a combination LMs associated to every document representation:

$$P(w|d) = \sum_i \lambda_i \cdot P(w|d_i) \quad (1)$$

where $P(w|d_i)$ is the i-th representation's LM, and λ_i is the weight on the i-th model, with $\sum_i \lambda_i = 1$. In our case, we build one LM for every field plus one LM for the document as a whole. This leads to a document model based on combining five different representations. Again, these models are smoothed with Dirichlet or Jelinek-Mercer smoothing.

5. EXPERIMENTS

We experimented with the four collections described in Section 4. For training purposes, for each collection, we generated 140 simulated queries and relevance judgments by taking account each of the variables of interest (10 queries for each of the 14 situations [see 2nd column in Table 2]). This provides a diverse training set and avoids over-fitting to any particular situation (the parameters are tuned from generic queries). Training was done by parameter sweeping[5], where we optimized for Mean Reciprocal Rank (MRR), which is a standard measure to evaluate known-item search algorithms. For simplicity, the smoothing configuration was fixed for each situation after training (i.e. the performance reported for the LMs refers to the LMs with the smoothing configuration, Dirichlet or Jelinek-Mercer, that was optimal in the training for the given situation).

Once the parameters for each model were fixed, the testing stage was done with new simulated queries (100 queries for each situation). To account for the randomness within the query simulation process we repeated the testing process 10 times (with 10 different 100-query sets) and we report the average MRR obtained. The collections were indexed with Indri and we removed 733 common words from the emails. No stemming was applied.

The results are presented in Table 3. For each situation, the values for the best performing model and all models that did not perform statistically poorer than the best are highlighted in bold[6].

Despite different query profiles being used, the results show that overall lm email and lmmix were the best performing models. In each of the investigated situations one of these models (and often both) was (were) in the the set of best performing models. For some situations the model bm25 was included in the set of best performing models. Overall, however, bm25 is not able to compete with lm email or lmmix. The models lm email and bm25 are similar in the sense that both of them represent the document as a whole. Still, the evolved term weighting incorporated within bm25 does not seem to offer added value wrt the simpler weighting schemes implemented by LM approaches [32]. This might be due to the characteristics of our retrieval task, where lengths of documents and term statistics deviate strongly from those found in more standard document retrieval scenarios.

Despite many of the queries consisting of single terms and many of these being NEs, the models that represent the emails with a single field (i.e. lm from, lm to, lm subject and lm body) perform poorly and are never as effective as the best models. Overall, the results clearly indicate that lm email or lmmix are optimal for re-finding email messages. Nevertheless, there are some trends in the results that, depending on the contextual situation, could help us decide which of these two models to use. For instance, if the collection is old then we should select lm email because it is always among the best performing models whereas lmmix is not. In constrast, for new or medium collection ages we should go for lmmix, which performs stronger overall than lm email in these contexts. Similar analyses lead us to conclude that a) lm email is preferable when the user does not use folders, while lmmix is better for spring cleaners; b) lm email performs more consistently for difficult tasks and lmmix is better choice for tasks with easy or medium difficulty, and c) lm email is a good choice for re-finding messages that have not been accessed for long time periods (cold temperature tasks) whereas lmmix is better for re-finding messages that have been accessed more recently (warm or hot temperature tasks).

Summing up, our findings reveal a quite interesting trend. It seems that when conditions are somehow difficult (the sought-after email has not been accessed for a long time, the task is perceived as difficult, users who do not file messages, or the collection is old) then we should go for a simple model that does not take into account structure and simply represents the document as a bag of words (lm email). In contrast, when conditions are somehow easier, then lmmix, which is a more evolved model that considers weights for the email fields, seems to be a more suitable choice.

We would also like to note that many of these scenarios can be automatically detected (e.g. the age of the collection or the user filing strategy) and, therefore, search applications can directly adapt their behaviour depending on the context. Some situations, such as the temperature of the task or the task difficulty, are more difficult to infer. Our findings seem to endorse further research on methods of automatically detecting such contextual variables. This could be achieved based on learning from the user interactions with the system including factors, such as the type and number of queries submitted, query sessions, etc.

6. LIMITATIONS AND DISCUSSION

The approach taken in this paper is extremely novel. We used user study data analyses to directly seed the query simulation process for evaluating retrieval models for email search. Ideally, however, before evaluating the performance of the models, we should try to establish the validity of the simulated queries. According to Zeigler [31], there are three kinds of validation that can be performed on a simulation; predictive, structural and replicative. A model has predictive validity if it can produce the same data output as the real system (i.e. comparing the query terms for a given known-item from the simulated model and a real system). A model has structural validity if the way it operates is a reflection of how the real system operates. Finally, replicative validity is achieved if the model produces output that

[5]For BM25, we fixed k_3 to 1000 (the effect of k_3 is negligible for short queries like ours) and varied k_1 (from 0 to 10 in steps of 0.2) and b (from 0 to 1 in steps of 0.1). For the LMs, we varied the smoothing parameters as follows: μ (Dirichlet smoothing parameter) from 0 to 5000 in steps of 500, and λ (Jelinek-Mercer smoothing parameter) from 0 to 1 in steps of 0.1. For lmmix we also varied the λ_i's between 0 and 1 in steps of 0.1

[6]Statistical significance was estimated with a paired t-test (p-value=0.05)

		Collection Age			User Experience		Filing Strategy			Task Difficulty			Task Temperature		
		new	medium	old	high	low	filers	no filers	spring	easy	medium	difficult	cold	warm	hot
CS	bm25	**0.438**	**0.430**	0.468	**0.472**	0.473	**0.468**	0.468	0.445	0.437	0.477	0.449	0.457	**0.480**	**0.450**
	lm from	0.221	0.264	0.213	0.247	0.267	0.239	0.216	0.233	0.239	0.285	0.156	0.187	0.283	0.210
	lm to	0.074	0.060	0.034	0.066	0.068	0.046	0.033	0.054	0.072	0.041	0.053	0.053	0.057	0.062
	lm subject	0.211	0.159	0.246	0.233	0.179	0.185	0.171	0.257	0.163	0.225	0.249	0.186	0.199	0.283
	lm body	0.278	0.295	0.330	0.304	0.289	0.336	0.333	0.299	0.281	0.308	0.321	0.331	0.322	0.296
	lm email	**0.439**	**0.433**	**0.479**	**0.471**	**0.481**	**0.474**	**0.484**	**0.455**	**0.446**	**0.493**	**0.459**	**0.468**	**0.484**	**0.454**
	lmmix	**0.439**	**0.433**	**0.479**	**0.471**	**0.481**	**0.474**	**0.484**	**0.455**	**0.446**	**0.493**	**0.459**	**0.468**	**0.484**	**0.454**
W3C-U1	bm25	0.201	0.175	0.262	0.231	0.197	0.206	0.188	0.203	0.209	0.203	0.219	0.199	0.186	0.262
	lm from	0.102	0.111	0.111	0.120	0.097	0.104	0.091	0.115	0.124	0.147	0.068	0.067	0.131	0.108
	lm to	0.072	0.055	0.054	0.081	0.082	0.053	0.045	0.068	0.075	0.045	0.073	0.057	0.051	0.092
	lm subject	0.126	0.098	0.162	0.137	0.101	0.115	0.091	0.147	0.095	0.138	0.159	0.112	0.118	0.173
	lm body	0.149	0.136	0.204	0.161	0.153	0.156	0.164	0.136	0.162	0.143	0.177	0.166	0.138	0.198
	lm email	0.219	0.201	**0.287**	0.249	**0.223**	**0.235**	**0.224**	0.223	0.238	0.236	**0.259**	**0.230**	0.204	0.303
	lmmix	**0.249**	**0.221**	**0.271**	**0.279**	**0.239**	**0.235**	**0.215**	**0.277**	**0.271**	**0.261**	0.237	0.209	**0.256**	**0.332**
W3C-U2	bm25	0.178	0.173	0.221	0.193	0.175	0.215	0.204	0.178	0.190	0.192	0.225	0.193	0.175	0.232
	lm from	0.054	0.071	0.047	0.069	0.076	0.067	0.052	0.072	0.069	0.078	0.043	0.048	0.061	0.061
	lm to	0.024	0.023	0.013	0.024	0.028	0.020	0.014	0.022	0.027	0.010	0.020	0.020	0.025	0.028
	lm subject	0.114	0.099	0.137	0.125	0.094	0.104	0.097	0.155	0.088	0.144	0.163	0.110	0.127	0.166
	lm body	0.111	0.121	0.165	0.120	0.122	0.157	0.172	0.107	0.142	0.124	0.179	0.152	0.129	0.136
	lm email	0.186	**0.182**	**0.249**	0.210	0.190	**0.232**	**0.216**	0.191	**0.204**	0.206	**0.241**	**0.212**	0.189	**0.240**
	lmmix	**0.199**	**0.194**	0.213	**0.224**	**0.206**	**0.217**	0.198	**0.219**	**0.219**	**0.237**	**0.232**	**0.211**	**0.207**	**0.251**
W3C-U3	bm25	0.205	0.209	0.270	0.251	0.230	0.219	0.229	0.235	0.222	0.241	0.246	0.203	0.216	0.259
	lm from	0.102	0.120	0.097	0.121	0.119	0.101	0.077	0.119	0.123	0.131	0.065	0.069	0.119	0.104
	lm to	0.042	0.015	0.025	0.046	0.020	0.035	0.023	0.021	0.026	0.030	0.029	0.019	0.028	0.040
	lm subject	0.144	0.096	0.177	0.150	0.122	0.129	0.093	0.173	0.110	0.149	0.174	0.114	0.136	0.188
	lm body	0.149	0.142	0.220	0.164	0.164	0.175	0.183	0.144	0.174	0.157	0.185	0.155	0.157	0.189
	lm email	**0.221**	**0.228**	**0.297**	**0.259**	**0.240**	**0.237**	**0.256**	**0.249**	**0.235**	**0.256**	**0.268**	**0.221**	**0.230**	**0.272**
	lmmix	**0.221**	**0.228**	**0.297**	**0.259**	**0.240**	**0.237**	**0.256**	**0.249**	**0.235**	**0.256**	**0.268**	**0.221**	**0.230**	**0.272**

Table 3: Mean Reciprocal Rank of the retrieval models in the 14 different situations. For each situation, the bolded figures refer either to the highest performing model or to those models whose difference wrt to the highest performing model is not statistically significant.

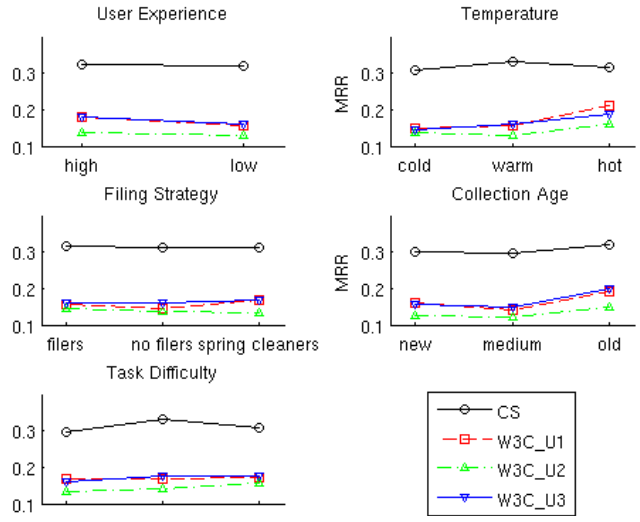

Figure 1: Mean performance of the retrieval models for different values of the contextual variables

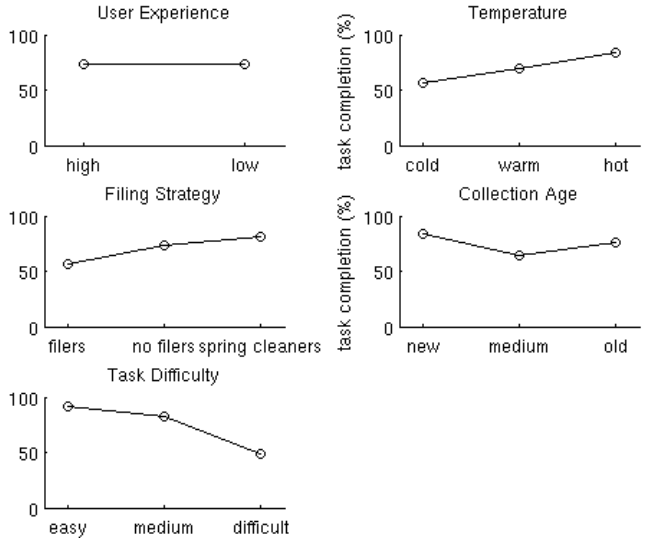

Figure 2: User Study Performance in terms of percentage of task completion.

is similar to the output of the real system (e.g. equivalent retrieval performance).

Previous work on query simulation for known-item tasks has established validity using both predictive [20] and replicative approaches [3, 20]. As our simulations build on the approaches used in these studies and in the case of [20] we make use of the same datasets, it is likely that the queries generated as a whole will be similarly valid. Nevertheless, it would be nice to have been able to validate our approach at the level of situation. Unfortunately, we are restricted in terms of what we can do to validate our simulated queries. We cannot perform predictive or replicative validity because these approaches require real queries along with an appropriate test collections.

In our case, we have real queries (from the user study) but do not have appropriate collection data (privacy concerns mean we cannot access user study participant collections). We also have publicly available test collections, but unfortunately for these collections we do not have suitable real queries against which to validate our simulated queries. What we can do with respect to replicative validity is compare the average performance of the evaluated models with the performance in the user study. Figure 1 shows the trends for the systems evaluation and Figure 2 provides the user study performance in terms of rate of task completion.

The results for the User experience and Task Temperature variables follow similar trends in both the system and user evaluations. This suggests that our simulations are most accurate in these situations. Also, for the filing strategy variable, in both evaluations, the spring-cleaners were the best performing group. The final two variables, Collection Age and Task Difficulty have trends that do not at all match the user study performance. It is important to note, however, that the user study results are user performance figures and do not necessarily reflect query performance. Although the quality of query will likely be a good indicator of overall performance, if the user is skilled at looking through lots of messages or in recognising relevant results this will skew the results.

It is also possible that for the Collection Age variable, the size of collection may be a larger factor in overall performance than the query generated. Older collections will most likely be larger in size and our results in Section 5 demonstrate how collection size can influence performance. The CS dataset (806 emails) was associated with much higher performance than the other collections (1892-3943 emails). This underlines the need for other ways of validating queries. It also suggests that bigger test collections are needed for this kind of work. The collections we used are equivalent in size to typical email collections [30, 13]. However, collections can be much bigger [18]. It is important that automated experiments reflect this.

There are other limitations to our work that we should mention. Although we extend previous simulations to incorporate fields and NEs, we do not consider how discriminative the selected terms or queries are. In the past, some effective simulations [3] have been based on selecting discriminative terms from the documents (tf/idf-like term selection). Nevertheless, our documents tend to be short, we remove common words and NEs make up a significant number of our query terms. Therefore, it is unlikely that further extending our simulations to consider the discriminative power of the terms would improve the realism of our simulation.

Another limitation of our work is that our simulation (as in [3, 20, 21]) only draws terms, named entities etc. from within the target document so the simulation does not account for user error. It would be relatively simple to include terms outside the document, for instance by interpolating terms with a collection language model, but we felt it would beyond the scope of this paper to do this. Similarly, we do not consider spelling mistakes or the fact that often multi-term queries will be phrases.

7. CONCLUSIONS AND FUTURE WORK

This paper has made two main contributions by first, investigating email re-finding queries and then second, by using the findings to seed a simulated evaluation of retrieval models for email search. The main findings can be summarised as follows:

- There was clear evidence in the user study of relationships existing between situational variables and query characteristics.
- We discovered that the age of the collection, the time that had elapsed since last accessing the email, the perceived difficulty of the task, the experience of the user, and the filing strategy the user adopted all had some influence on one or more of the following characteristics: the length of the query, the field submitted to, and the use of named-entities.
- We incorporated the trends as they appeared in the user study data in a simulated evaluation of retrieval models for email search.
- We learned about the performance of various models for email search. The results showed that models that make use of the whole document (lm email and lmmix and sometimes bm25) achieved the best performance.
- We uncovered situations when it is more sensible to make use of structural information of the document. A pattern emerged indicating that in more difficult situations it best to use a simple model that ignores structure, whereas in easier situations better performance can be achieved by taking structure into account.

Although we feel that the presented work is an important starting point for the fusion of user and systems IR approaches, it is important to acknowledge that it is only the starting point. Currently we are working on ways to improve the methodology in order to incorporate a means of validating simulated queries. We are also looking to extend the approach to look at other kinds of personal data including visited web pages, personal files, and calender entries. In other related work we are looking at ways of implicitly detecting contextual variables from live re-finding behaviour so that search applications can make use of different retrieval models appropriately depending on the contextual situation.

Acknowledgments

The first author was supported by the Alexander von Humboldt Foundation, Germany. The last three authors thank the financial support given by Ministerio de Ciencia e Innovación through research project ref. TIN2010-18552-C03-03. We would also like to thank Jinyoung Kim for providing us with the test collections.

8. REFERENCES

[1] A. Aula, N. Jhaveri, and M. Käki, *Information search and reaccess strategies of experienced web users*, Proc. Int. Conf. World Wide Web, 2005, pp. 583–592.

[2] A. Aula, R. M. Khan, and Z. Guan, *How does search behavior change as search becomes more difficult?*, Proc. SIGCHI conference on Human factors in computing systems, 2010, pp. 35–44.

[3] L. Azzopardi, M. de Rijke, and K. Balog, *Building simulated queries for known-item topics: an analysis using six european languages*, Proc. ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 455–462.

[4] R. Boardman and M. A. Sasse, *"stuff goes into the computer and doesn't come out": a cross-tool study of personal information management*, CHI '04: Proc. SIGCHI, 2004, pp. 583–590.

[5] H. Bruce, W. Jones, and S. Dumais, *Keeping and re-finding information on the web: What do people do and what do they need?*, ASIST 2004: Proceedings of the 67th ASIST annual meeting, October 2004.

[6] R. G. Capra and M. A. Perez-Quinones, *Re-finding found things: An exploratory study of how users re-find information*, Tech. report, Computer Science Dept., Virginia Tech, 2003.

[7] R. G. Capra and M. A. Perez-Quinones, *Using web search engines to find and refind information*, Computer **38** (2005), no. 10, 36–42.

[8] S. Chernov, P. Serdyukov, P. Chirita, G. Demartini, and W. Nejdl, *Building a desktop search test-bed*, ECIR'07: Proceedings of the 29th European conference on IR research (Berlin, Heidelberg), Springer-Verlag, 2007, pp. 686–690.

[9] S. Cohen, C. Domshlak, and N. Zwerdling, *On ranking techniques for desktop search*, ACM Trans. Inf. Syst. **26** (2008), 11:1–11:24.

[10] E. Cutrell, D. Robbins, S. Dumais, and R. Sarin, *Fast, flexible filtering with phlat*, Proc. SIGCHI conference on Human Factors in computing systems, 2006, pp. 261–270.

[11] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D.C. Robbins, *Stuff i've seen: a system for personal information retrieval and re-use*, Proc. ACM SIGIR '03:, 2003, pp. 72–79.

[12] D. Elsweiler, *Supporting human memory in personal information management*, Ph.D. thesis, Department of Computer and Information Sciences, University of Strathclyde, 2007.

[13] D. Elsweiler, M. Baillie, and I. Ruthven, *Exploring memory in email refinding*, ACM Trans. Inf. Syst. **26** (2008), no. 4, 1–36.

[14] D. Elsweiler, M. Baillie. and I. Ruthven, *On understanding the relationship between recollection and refinding*, Journal of Digital Information (JoDI) **10** (2009), no. 5.

[15] D. Elsweiler, M. Baillie, and I. Ruthven, *What makes re-finding information diffcult? a study of email re-finding*, Proc. ECIR 2011, 2011.

[16] D. Elsweiler, G. Jones, L. Kelly, and J. Teevan, *Workshop on desktop search*, SIGIR Forum **44** (2010), 28–34.

[17] D. Elsweiler and I. Ruthven, *Towards task-based personal information management evaluations*, Proc. ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 23–30.

[18] D. Fisher, A. J. Brush, E. Gleave, and M. A. Smith, *Revisiting whittaker & sidner's "email overload" ten years later*, Proc. Conference on Computer supported cooperative work, 2006, pp. 309–312.

[19] D. Kelly and J. Teevan, *Personal information management*, ch. Understanding what works: Evaluating personal information management tools, pp. 190–204, Seattle: University of Washington Press., 2007.

[20] J. Kim and W. B. Croft, *Retrieval experiments using pseudo-desktop collections*, CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management (New York, NY, USA), ACM, 2009, pp. 1297–1306.

[21] J. Kim and W. B. Croft, *Ranking using multiple document types in desktop search*, Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval (New York, NY, USA), SIGIR '10, ACM, 2010, pp. 50–57.

[22] P. McCullagh and J. A. Nelder, *Generalized linear models (second edition)*, Chapman and Hall, 1989.

[23] B. McKenzie and A. Cockburn, *An empirical analysis of web page revisitation*, Proc. of the HICSS34, 2001.

[24] P. Ogilvie and J. Callan, *Combining document representations for known item search*, Proc. ACM SIGIR '03:, 2003, pp. 143–150.

[25] M. Ringel, E. Cutrell, S. Dumais, and E. Horvitz, *Milestones in time: The value of landmarks in retrieving information from personal stores.*, Proc. INTERACT 2003, 2003, pp. 184–191.

[26] A. J. Sellen and R. H. R. Harper, *The myth of the paperless office*, MIT Press, Cambridge, MA, USA, 2003.

[27] J. Teevan, E. Adar, R. Jones, and M. Potts, *Information re-retrieval: Repeat queries in yahoo's logs*, Proc. SIGIR '07, 2007.

[28] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger, *The perfect search engine is not enough: a study of*

orienteering behavior in directed search, Proc. SIGCHI conference on Human factors in computing systems, 2004, pp. 415–422.

[29] S. K. Tyler and J. Teevan, *Large scale query log analysis of re-finding*, Proc. WSDM '10, 2010.

[30] S. Whittaker and C. Sidner, *Email overload: exploring personal information management of email*, Proc. SIGCHI conference on Human factors in computing systems, 1996, pp. 276–283.

[31] Bernard P. Zeigler, *Theory of modelling and simulation*, Krieger Publishing Co., Inc., Melbourne, FL, USA, 1984.

[32] C. Zhai and J. Lafferty, *A study of smoothing methods for language models applied to information retrieval*, ACM Transactions on Information Systems **22** (2004), no. 2, 179–214.

APPENDIX

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	-1.418e+00	9.404e-01	-1.508	0.132622
Collection Size	-4.007e-05	2.727e-05	-1.469	0.142716
number_folders	3.580e-03	1.886e-03	1.898	0.058533
Avg.#emails/day	**3.351e-02**	**1.366e-02**	**2.452**	**<0.05**
User experience:low	**1.339e-01**	**5.751e-02**	**2.329**	**<0.05**
Difficulty:hard	**1.705e+00**	**8.590e-01**	**1.985**	**<0.05**
Difficulty:easy	**1.784e+00**	**8.591e-01**	**2.077**	**<0.05**
Difficulty:medium	**1.880e+00**	**8.528e-01**	**2.205**	**<0.05**
When	**2.405e-01**	**5.535e-02**	**4.345**	**<0.01**
Sender	**2.203e-01**	**5.926e-02**	**3.718**	**<0.01**
Topic	**-3.110e-01**	**7.839e-02**	**-3.968**	**<0.01**

Table 4: Regression Model for hasSender

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	2.283e+00	1.139e+00	2.004	<0.05
Collection size	1.055e-05	1.696e-05	0.622	0.53421
number_folders	7.031e-04	2.190e-03	0.321	0.74843
Collection age:new	**2.563e-01**	**1.009e-01**	**2.539**	**<0.05**
Collection age:old	**1.890e-01**	**8.176e-02**	**2.311**	**<0.05**
User experience:low	**-2.508e-01**	**8.440e-02**	**-2.971**	**<0.01**
Task-type:item	-9.696e-02	7.429e-02	-1.305	0.19273
Task-type:multi	1.306e-01	9.198e-02	1.420	0.15654
Filing:no filer	-4.944e-02	9.931e-02	-0.498	0.61893
Filing:spring	-1.219e-01	1.142e-01	-1.067	0.28659
pre_contains:not sure	-3.169e-01	1.741e-01	-1.820	0.06967
pre_contains:yes	-2.739e-01	1.462e-01	-1.874	0.06186
Temperature:hot	1.653e-01	9.604e-02	1.721	0.08623
Temperature:Range	-2.481e-01	1.129e-01	-2.198	**<0.05**
Temperature:warm	2.453e-02	8.752e-02	0.280	0.77945
Task Freq:infreq.	7.729e-03	7.004e-02	0.110	0.91220
Task Freq:freq.	-4.040e-02	8.853e-02	-0.456	0.64845
Task difficulty:hard	-1.532e+00	1.068e+00	-1.435	0.15216
Task difficulty:easy	-1.510e+00	1.061e+00	-1.423	0.15567
Task difficulty:medium	-1.596e+00	1.056e+00	-1.510	0.13192
When	-2.291e-02	6.678e-02	-0.343	0.73180
Sender	-2.184e-01	6.714e-02	-3.253	**<0.01**
Topic	8.805e-02	1.111e-01	0.793	0.42846
Reason	-4.873e-02	9.431e-02	-0.517	0.60570
Other.recp	-4.206e-02	6.660e-02	-0.632	0.52816

Table 5: Regression Model for hasSubject

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	0.619595	0.139081	4.455	<0.01
Avg.#emails/day	**-0.024198**	**0.005725**	**-4.227**	**<0.01**
number_folders	**0.003369**	**0.001199**	**2.810**	**<0.01**
Filing:no filer	**0.158860**	**0.072745**	**2.184**	**<0.05**
Filing:spring	**-0.152566**	**0.068970**	**-2.212**	**<0.05**
pre_contains:not sure	-0.269388	0.146071	-1.844	0.066029
pre_contains:yes	-0.114047	0.124052	-0.919	0.358573
Temperature:hot	**-0.276521**	**0.064366**	**-4.296**	**<0.01**
Temperature:range	**-0.264592**	**0.068707**	**-3.851**	**<0.01**
Temperature:warm	**-0.361725**	**0.064131**	**-5.640**	**<0.01**
When	**-0.095110**	**0.030288**	**-3.140**	**<0.01**

Table 6: Regression Model for hasBody

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	c	0.47430	0.07363	6.442 <0.01
Collection age:new	-0.02863	0.06374	-0.449	0.65360
Collection age:old	**0.19052**	**0.07288**	**2.614**	**<0.01**
Filing:no filer	0.15413	0.08061	1.912	0.05669
Filing:spring	0.02966	0.08650	0.343	0.73185

Table 7: Regression Model for hasNamedEntity

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	8.077e-01	3.529e-01	2.289	<0.05
Collection size	-1.674e-05	1.142e-05	-1.465	0.1437
Collection age:new	-6.605e-02	7.705e-02	-0.857	0.3919
Collection age:old	1.283e-01	7.260e-02	1.768	0.0780
User experience:low	1.253e-01	6.630e-02	1.889	0.0597
pre_contains:not sure	1.777e-01	1.708e-01	1.040	0.2989
pre_contains:yes	2.788e-01	1.519e-01	1.836	0.0673
Temperature:hot	**-1.730e-01**	**7.830e-02**	**-2.210**	**<0.05**
Temperature:range	-4.201e-03	8.737e-02	-0.048	0.9617
Temperature:warm	1.339e-02	7.903e-02	0.169	0.8656
Task difficulty:hard	-7.429e-01	3.805e-01	-1.952	0.0517
Task difficulty:easy	-5.788e-01	3.763e-01	-1.538	0.1249
Task difficulty:medium	-6.862e-01	3.813e-01	-1.800	0.0728

Table 8: Regression Model for hasPerson

	Est.	Std.Error	t value	Pr(> |t|)
(Intercept)	0.067843	0.079382	0.855	0.393358
number_folders	0.001858	0.001143	1.625	0.105018
User experience:low	-0.094082	0.048853	-1.926	0.054966
Filing:no filer	**0.208101**	**0.075727**	**2.748**	**<0.01**
Filing:spring	0.004585	0.076354	0.060	0.952148
Temperature:hot	**0.238175**	**0.061590**	**3.867**	**<0.01**
Temperature:range	-0.035453	0.063167	-0.561	0.574990
Temperature:warm	0.098244	0.062560	1.570	0.117258

Table 9: Regression Model for hasOtherNamedEntity