Evaluation of context-aware recommendation systems for information re-finding

Maya Sappelli [a,b], *Suzan Verberne* [b], *Wessel Kraaij* [a,b]

Abstract

In this paper we evaluate context-aware recommendation systems for information re-finding that observe knowledge workers in their daily worktasks. The agents interpret the interaction of users with their regular office PC. The recognized activities are used to recommend information in order to let the user work focused and efficiently.

In order to decide which recommendation method is most beneficial to the knowledge worker, it is important to determine in what way the knowledge worker should be supported. From the knowledge worker scenario we identify four evaluation criteria that are relevant for evaluating the quality of knowledge worker support: context relevance, document relevance, prediction of user action and diversity of the suggestions.

We compare three different context-aware recommendation methods for information re-finding in a task setting where the agent provides the user with document suggestions that support their active (writing) task. Each method uses a different approach to context-awareness. The first method uses contextual pre-filtering in combination with content based recommendation (CBR), the second uses the just-in-time information retrieval paradigm (JITIR) and the third is a novel network-based recommendation system where context is part of the recommendation model (CIA). These methods are also compared to a random baseline.

We found that each method has its own strengths: CBR is strong at context relevance, JITIR captures document relevance well and CIA achieves the best result at predicting user action. Weaknesses include that CBR depends on a manual source to determine the context and the context query in JITIR can fail when the textual content is not sufficient.

We conclude that to truly support a knowledge worker, all four evaluation criteria are important. In light of that conclusion, we argue that the network-

a) Media and Networking Services, TNO, Anna van Buerenplein 1, 2595 DA Den Haag, The Netherlands

b) Institute for Computing and Information Sciences, Radboud University Nijmegen, Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands

E-mail: m.sappelli@cs.ru.nl

based approach CIA offers the highest robustness and flexibility for context-aware information recommendation.

Keywords context-aware document recommendation, re-finding, just in time information retrieval, recommender systems, evaluation

1 Introduction

Many knowledge workers who process and produce information are confronted with a phenomenon referred to as `information overload' (Bawden and Robinson, 2009). Knowledge workers can get overwhelmed by the amount of information they need to handle. In our project SWELL[1] we aim to support these knowledge workers in their daily working life by helping them to `work in context' (Gomez-Perez et al., 2009; Warren, 2013). This means that we try to help the user stay focused on his tasks by recommending documents from the user's work history (documents and webpages that the user has previously accessed) that are relevant to his current working context (i.e. current activities and topics). This paper describes how we can evaluate context-aware document recommendation systems for knowledge worker support in a document re-finding setting, and provides an evaluation of three approaches to context awareness in document recommendation.

It has been shown that people often forget to use documents that can be helpful, even when they are stored in an appropriate location (Elsweiler et al., 2007). The recommendation of documents can improve task performance by the reduction of the number of computer interactions required, and has been showed to improve the perceived usability of an information re-finding system (Wakeling et al., 2014). Especially for the task of writing, the time to complete the task can be shortened and the quality of the written document can be improved when relevant information is pro-actively recommended (Melguizo et al., 2010). This suggests that a recommender system for re-finding information can be useful for a knowledge worker. The time to complete a task or the quality of a written document would be the perfect extrinsic evaluation criteria for the evaluation of a recommender system for re-finding information. However, this data is not available and costly to obtain. Therefore we investigated the potential of using a pre-existing offline knowledge worker dataset (Sappelli et al., 2014) for the evaluation of simulated context-aware document re-finding.

We argue that there are several ways in which a context-aware recommendation system for information re-finding can be useful to a knowledge worker. For that purpose we describe a knowledge worker scenario. Four evaluation criteria are derived from the knowledge worker scenario, each with their own evaluation metrics. Ideally, a good system would score well on all evaluation criteria. We evaluate three approaches to context-aware information recommendation on each of these criteria. Our research questions are:

1. How should we evaluate a context-aware information recommendation system in light of the goal to support knowledge workers in re-finding information?
2. What are the benefits and downsides of content-based recommendation with pre-filtering, just-in-time information retrieval, and context-modelling as methods for recommending documents with the purpose of helping the knowledge worker?

[1] http://www.swell-project.net

We focus on supporting the knowledge worker through document recommendation, which is why we present a discussion of related work on document recommendation, just-in-time information retrieval and context-aware recommendation in Section 3. In Section 4 we present the four criteria of evaluation which are derived from the knowledge worker scenario. This is followed by an experiment in which we compare the effectiveness of three methods for incorporating context for recommending documents in a knowledge worker setting. These methods are described in Section 5 and the results of the experiment in Section 6.

2 The knowledge worker scenario

A knowledge worker is a person that works mainly with information. He uses and produces information. In our scenario we focus on knowledge workers who work mainly on a computer and process and produce information from documents in order to gain new knowledge.

A typical workday of such a knowledge worker can be described by a combination of activities. Some activities are organizational in nature, such as handling e-mail messages or attending meetings. Some activities are more substantial, such as writing project proposals or reports and preparing presentations. Depending on the type of knowledge worker, software programming or data analysis can also be part of the job.

Consider Bob, he is a 43 year old programmer at a large company. He starts his day with finishing up a report on his latest Java deep-learning project. Only a couple of details and references are needed, but he needs to finish this work before 1 pm. He knows that the papers he needs as references in his report are somewhere on his computer, because he has read them before. At this point he could be helped by opening these documents for him, to spare him the time to navigate to them or look for them himself.

At 11 am he realizes that he is missing a piece of information. He has read it before, but cannot remember where and starts to search on his computer. Bob finds some information about deep-learning in Python, which he also saved on his computer. Because Python is relatively new to him, he finds it more interesting than his current Java project and he gets distracted. At 12.30 he realizes that he has spent too much time learning about deep-learning in Python and that he only has 30 minutes left to finish his project. He finishes it quickly.

In the meantime a couple of e-mail messages have arrived for Bob. Most of them are not so important, but one is about the possibility to work on new, self-defined research. Bob has wanted this for a while, so decides to write a proposal. He already has an idea about the topic he wants to pursue, but he wants to challenge himself. At this point Bob could be helped by thinking out of the box, and suggesting him documents that are related to the topic, but cover a variety of perspectives.

At 5 pm Bob finishes his day. He has found so many documents for his new project proposal that he feels a little bit overwhelmed. He has not been able to read all documents yet. He decides to catch up on some reading at home.

Our aim is to support Bob in his information management. We see four ways to support him:

(a) By preventing distractions for the knowledge worker so that he can finish his task effectively.
(b) By reminding the knowledge worker of information that he has seen before and is relevant now.
(c) By pre-fetching the documents that he needs for the current task, so that he saves time in navigating to them.
(d) By providing a diverse range of items to spark the knowledge worker's creativity when he needs it.

These four support methods are the foundation for the evaluation criteria that we use to evaluate the context-aware recommendation systems for information re-finding.

3 Related work

In this section we describe previous work related to the research in this paper. Our work relates to several areas of research: information retrieval, recommender systems (Ricci et al., 2011), information behaviour in context (Ingwersen and Järvelin, 2006) and user-centric evaluation of information systems (Kelly, 2009). In this section we restrict ourselves to related work on a) system-initiated methods for document recommendation (i.e. no search systems) in Sections 3.1 and 3.3, and b) context-aware methods in Section 3.2. In terms of evaluation we focus on online evaluation methods, which are described in Section 4

3.1 Document Recommendation

There are several traditional recommendation approaches to provide users with documents during their work. Most of these make use of collaborative filtering techniques to find relevant documents. Weng and Chang (2008) construct a user profile ontology to reason about the interests of users. They search for user groups with similar interests using a spreading activation model and use their interests as basis for the recommendations of new documents.

In another approach, Lakiotaki et al. (2011) model the recommendation problem as a decision problem (which document should I use next?), and investigate the use of multiple-criteria decision analysis (MCDA) as method for user profile construction. The authors conclude that MCDA and the subsequent clustering of these profiles enhances the performance collaborative-filtering techniques.

More recently, Lai et al. (2013) have taken the trustworthiness of the ratings by users into account. They propose several methods that use both personal trust as well as group trust. Their proposed methods had lower mean average errors than methods that do not take trustworthiness into account and methods that only use user trustworthiness. This was evaluated on a dataset from a knowledge management system consisting of 800 documents and 80 knowledge workers with their access and rating behaviour.

Although these methods are valuable, they only consider the user and his (relatively static) interests, and not the user's current working context. The recommendations are aimed to be of general interest to the user. These interesting items are not necessarily useful at the time they are recommended and can be a potential source of distraction. Our goal is to reduce the information overload of a user, not to add to it. For this purpose it is important to look at what the user needs, rather than what the user might like. To avoid overload we focus on re-finding information, therefore the user's needs will be to re-find information sources that are relevant for his current activities. Typically, re-finding involves the user issuing queries (Dumais et al., 2003), but in this paper the focus is on proactive recommendation of documents that the user has seen before.

The task of re-finding information is strongly related to memory (Elsweiler et al., 2007). This has lead to the hypothesis that contextual elements can also help people to re-find items. Blanc-Brude and Scapin (2007) investigated what people recall about documents they have seen and what this implies for search tools. They found that the aspects of documents that users often recall are contextual elements such as keywords, file location, file type, document format, time of last usage, associated events and visual elements. In addition, Kelly et al. (2008) conclude that as the recall of the content itself declines, contextual information becomes more important to re-find information.

More recently, Chen and Jones (2014) have investigated the usefulness of episodic context in a search system for re-finding information. They describe experiments in which they assess the episodic features people remember, which include the name of desktop applications and websites, the name and contact of an e-mail and the information that represent the content of the activity. Although the episodic or contextual features were not frequently used in queries, they did improve the effectiveness of re-finding tasks. File extension, contact names and temporal information were most often used as contextual attributes to a query.

By using the current context of a user, we can find documents that have a relation to a similar context. Since the context of document access, such as the person to which a document was sent or the day it was accessed, can be used to more effectively re-find a document, it is likely that a list of documents related to the user's current context contains documents that the user would potentially want to re-find. This means that for the task of re-finding information we should look at recommendation systems that take context into account. In the next section we describe such recommendation agents that take the user's context into account.

3.2 Context-aware recommendation and re-finding agents

There are roughly three methods to incorporate context in a recommendation agent: contextual pre-filtering, contextual post-filtering and contextual modelling (Ricci et al., 2011).

In the paradigm of contextual pre-filtering, the set of data that the recommender system uses is filtered for the context that is currently active. This means that simply all the possible suggestions that are not relevant for the current context are taken out before the ranking is determined. Typically the context in these kind of systems is some kind of category. For example a context for a movie recommendation system can be `watch with family' or `watch with friends'.

Pre-selection of contexts can be done by using the context as a query. For example, Sappelli et al. (2013) use the physical location of a user as query and rank the resulting potential tourist activities according to the user's preferences. When this pre-selection is too strict (e.g. there are too few search results for this context), context generalization can be applied (Ricci et al., 2011). In the tourist recommender system, this can be achieved by using a city as location query, rather than the exact GPS location.

A second method for incorporating context in a recommender system is contextual post-filtering. This is very similar to the pre-filtering case, but here the system produces a ranked list for all items, first ignoring any contextual information. The ranked list is re-ranked or filtered afterwards based on the context of interest (Ricci et al., 2011).

There is a last type of context-aware recommendation system where the context is part of the recommendation model. Oku et al. (2006) propose a contextual version of SVM where context-axes are incorporated in the feature space. Incorporating context using factorization methods is also popular (Karatzoglou et al., 2010; Rendle et al., 2011).

The downside of these methods is that the detection of what context is active is often not incorporated in the model. Typically the user is asked to select the context for his search. For example, he can select that he is watching a movie with friends tonight. This means though that all possible contexts need to be determined beforehand, and no personal contexts can be taken into account.

From the perspective of the knowledge worker, his most important context is the (topic of) the task he is working on. As the activities vary throughout the day, it would cost the knowledge worker much effort if he would have to indicate this each time he changes activities. This would diminish the possible advantages of using a recommendation system.

Additionally, reducing the context of a knowledge worker to fixed categories is a limitation, as slight variations in topics would not be captured. A more realistic and content-rich context of a knowledge worker would be the text of a (web) document he is observing at that moment.

3.3 Just-in-time information Retrieval

A special type of context-aware recommendation agents are the agents for just-in-time information retrieval (JITIR). In this setting, the context is used as a query in a search system. The system is pro-active in the sense that the querying takes place in the background, and the search results are presented to the user. Thus, the user does not need to select his context, which is an advantage over the context-aware recommendation agents described in the previous section. The context query that is used can be formulated from the document a person is writing (Budzik and Hammond, 2000; Melguizo et al., 2010), the blogpost he is writing (Gao and Bridge, 2010), e-mail messages that are being read or composed (Dumais et al., 2004), the news that is being broadcasted (Henzinger et al., 2005) or the text that is visible on screen together with the location, person, date and subject information (Rhodes, 1997).

A limitation of the JITIR agents is that the information leading up to the current context is ignored. The session information can contain valuable information

about what has already been seen and what not. Historic behaviour of users has proven to benefit personalized re-ranking of documents (Cai et al., 2014).

4 Evaluation for context-aware information recommendation

Ideally a context-aware information recommendation system for re-finding would be evaluated in an online interactive setting with users. In such a case, each user would work as he normally does, while receiving suggestions from one of the systems that is being evaluated. During the experiment we could evaluate whether the suggestions lead to improved task execution in terms of time profit or quality. Moreover, the user could be asked to rate the suggestions he receives at a certain moment. This method of evaluation, however, is expensive. Each system, or even each adaptation in system settings, would require a new period of evaluation with users. Furthermore, the extrinsic evaluation methods are not trivial: to assess time profit or quality of work, the tasks that are being evaluated should be equal. However, if a person executes the same tasks multiple times, there is a learning effect that should not be confused with the effect of using the system. Moreover, asking a user to provide ratings of the suggested documents during the experiment could influence the subsequent suggestions as they are dependent on what is happening on the user's screen, while rating the suggestions afterwards would make the ratings not context-dependent.

To overcome the issues of interactive evaluation, we opt to do an offline evaluation instead. For this purpose we define several criteria that a good context-aware document suggestion should meet. The criteria are motivated by the methods in which we can support knowledge workers as described in the knowledge worker scenario (Section 2). We use a dataset of knowledge worker activities (Sappelli et al., 2014) to simulate the work session of a knowledge worker, which enables us to evaluate the recommendations in a context-dependent setting. The assumptions underlying this approach do limit the generalisability of the conclusions. On the other hand, however, this offline way of evaluation has the advantage that the impact of small changes in system settings can be evaluated more easily. Moreover, it provides the possibility to reproduce results and provides a baseline for comparison for new systems.

There are multiple existing methods for the offline evaluation of (non context-dependent) recommender systems. Therefore we describe some standard evaluation practices for recommender systems in the remainder of this section. This is followed by the evaluation criteria that we have derived from the knowledge worker scenario (Section 2)

4.1 Standard evaluation practices for recommender systems

In the offline evaluation of recommender systems, the most important measure is predictive-based (Ricci et al., 2011). The assumption is that a system with more accurate predictions of what the user will do will be preferred by the users. There are two interpretations of predictive accuracy in recommender systems. In the first interpretation the system tries to predict the user's rating of an item. Mostly this

form of evaluation measures how well a system is capable of predicting how an item will be rated (e.g. movie ratings).

The second interpretation of predictive accuracy focuses on what the user will do with a suggestion. In this interpretation the evaluation focuses on how well a system can predict the action of a user. In a movie recommendation example this would focus not on how the user rates a movie, but on whether the user will actually buy or watch the suggested movie. Both aspects of predictive accuracy are useful to find documents that can support the knowledge worker. For example, we can predict whether a document contains relevant information, or we can predict which document a user will open next.

The case of the knowledge worker is not completely comparable to most recommendation systems. In terms of evaluation, the occurrence of false positives has a larger impact in knowledge worker support than in other recommendation systems such as movie recommendation. In movie recommendation, a bad recommendation will only have a small impact on the overall opinion about the system as long as there are not too many bad recommendations. In the case of the knowledge worker, a bad recommendation can distract the worker and disrupt his work ow, something that is diametrically opposed to the reason for using the recommendation system in the first place. This means that preventing distracting suggestions is an important aspect in the knowledge worker scenario.

We address four possibilities to support knowledge workers in the case of re-finding information, connected to the support options (a)--(d) from the knowledge worker scenario:

- **Context Relevance:** A knowledge worker can be supported by suggesting him documents that fit the topic of his current activities and therefore do not distract him
- **Document Relevance:** A knowledge worker can be supported by suggesting him documents that contain relevant information for the (writing) task he is working on. Where context relevance evaluates whether there is a general topical match with the current activities, document relevance is aimed at a more detailed evaluation of how much a suggested document contributes to the writing process.
- **Action Prediction***:* A knowledge worker can be supported by suggesting him documents that he is going to open in the near future
- **Diversity:** A knowledge worker can be supported by suggesting him a variety of documents

Each of these support possibilities can be seen as a criterion for evaluation. Each evaluation criterion has its own evaluation metric. We have chosen evaluation metrics for each of these criteria based on literature and availability of data. Therefore we start with a description of the data that is available to us, and then describe the evaluation metrics for each criterion.

4.2 Data

For the experiments described in this paper we make use of a publicly available dataset collected during a knowledge worker experiment (Sappelli et al., 2014). To our knowledge this is the only public dataset with comprehensive computer

interaction data that captures the context of knowledge workers realistically and without privacy issues. The interaction data allows for the simulation of a work session, in order to evaluate the context-aware recommendation process.

The dataset was collected during an experiment in which 25 participants were observed while executing typical knowledge worker tasks. The participants were asked to write reports on a total of 6 given topics and prepare presentations for three of the topics. The topics were `Stress at Work', `Healthy Living', `Privacy on the internet', `Tourist Attractions in Perth', `Road trip in USA', and `The life of Napoleon'. So, for each participant we have 6 written documents and 3 presentations that were produced for the task. In addition, we stored all (local and web) documents that were accessed by the users during their work session.

The data were collected in three sessions that mirror the knowledge worker scenario. Each session lasted between 30 and 45 minutes. The conditions were: a) a neutral session in which the participants were asked to work as they normally do; b) a session in which they were time pressured and c) a session in which they were interrupted with email messages. Some of these messages contained a task for the participant, which resulted in two additional topics in the data: `Einstein' and `Information Overload'.

The dataset that resulted from this experiment contains all computer interaction data that was recorded during the experiment. Most importantly the dataset contains the data originating from the uLog key logger[2] as well as browser history data collected with IEHistoryView.

For the experiments described in this paper we make use of the preprocessed version of the dataset. In this version of the dataset, individual events are aggregated to meaningful event blocks. The start of a new event block is defined as either an application switch event, or a change in window title event. All the individual events, such as the keys typed, and all captions (mouse-over tool tips), that occurred between application or window switches are concatenated into strings and the number of mouse clicks per event block is counted. From the recorded Google URLs the queries that were entered were extracted using a regular expression. In total, the data collection consists of 9416 event blocks with an average of 377 event blocks per participant. The average duration of an event block is 51.5 seconds. The average number of accessed documents per participant per 3-hour work session was 44 documents.

Table 1 shows an overview of the features collected per event block, with an example value for each feature.

4.2.1 Data Labelling

Table 1 shows that each event block was labelled with a topic label. This was done as a second step after the data collection experiment using the crowd sourcing platform Amazon Mechanical Turk. The event blocks were presented to the annotators in a desktop-like setting to mimic the desktop view of the user during the experiment. The annotators were asked to select 1 topic label and also indicate on a scale of 1-5 how certain they were of their decision. The event blocks were shown in random order, so they could not use any session information. The labels were the 8 topics (`Stress at Work', `Healthy Living', etc.), and an additional topic

[2] http://www.noldus.com/human-behaviour-research/products/ulog

Table 1 Overview of features collected per event block, with example values

Feature	example value
Id	6
participant id	2
begin time	20120919T132206577
end time	20120919T132229650
duration (seconds)	24
# clicks	3
typed keys	we austra;i lia
application	Iexplore
window title	Google - Windows Internet Explorer
caption	New Tab (Ctrl+T) New Tab (Ctrl+T)
url	http://www.google.nl/search?hl=nl& sclient=psy-ab&q=australia+&oq=australi
Domain	www.google.nl
Query	Australia
Label	Perth

`indeterminable' when the event block did not contain any identifiable topic, for example when just the website `www.google.nl' was shown.

Each document that was opened during the experiment was labelled with the topic label that was assigned to the event-block in which the document was accessed. A document can have multiple topic labels. In total there were 799 documents accessed during the experiment, of which 349 were tagged with the label `indeterminable'. We assume that within one event-block, a single topic guided the information access behaviour of the user. Table 2 presents the distribution of documents over the topic labels. Overall there were 43 documents that were associated with more than one topic. An example is http://healthypattern.com/things-you-can-do-at-work-to-relieve-stress-at-work.html which is associated with the topics Healthy Living and Stress. Some of these documents have multiple labels because of errors in the labelling by Amazon Mechanical Turk. An example of such a document is http://www.perthtourism.com.au/recreation.html which is associated with both Roadtrip and Perth. The roadtrip topic was about a roadtrip in the USA, so a website on Perth should not have been tagged with this label. We have not corrected these erroneous labels, as this kind of noise would occur in a live system as well.

Table 2 Overview of documents per topic

Topic	Number of documents
Einstein	19
Privacy	30
Information Overload	13
Roadtrip	138
Perth	127
Healthy Living	70
Stress	58
Napoleon	88

4.2.2 Using the data for recommendation

For the evaluation of context-aware re-finding we assume that the user is writing a document or preparing a presentation, similar as in this dataset. For the simulation of the re-finding task, we need a set of documents that the user has accessed before (in reality maybe weeks or months earlier), either stored locally on his computer or visited in his browser. The set of documents with a label other than `indeterminable' that are accessed during the experiments is on average 44 documents per participant, which is too small to evaluate a typical knowledge worker setting. Therefore, we extended the list of candidate documents with the documents accessed by all users. This is a set of 450 documents of which 95% are web-documents defined by an URL. The dataset shows that on average a knowledge worker accesses 18 documents per hour, thus we argue that the set of 450 documents represents a history of at least 25 hours of concentrated work. In reality this would be equivalent to a working week, since a normal working day also includes other activities such as meetings etc. We argue that the set of 450 documents is large enough to introduce re-finding problems. For each participant the work session is simulated by re-running the logged event blocks. For each event-block we determine the relevancy of each of the documents in the collection, rank them and select the top 10 as our recommendation list. This is motivated by the length of a typical search result page (10 search results). However, the optimal number of suggestions in context-aware document recommendation is an open topic for research that is not within the scope of this paper.

Documents that are open in the current event-block or have been opened in previous event-blocks in the current work session are filtered. The assumption is that documents that have been seen in the current work session should not be recommended because the user does not need help re-finding those. We assume that a session consists of the activities that are executed between system boot and system shut down, with a maximum duration of one day. The expectation is that documents that are accessed during a day are remembered by the user and do not need to be recommended. In the dataset the session of a participant equals the three-hour experiment in which he participated.

The recommendation lists are evaluated on the four knowledge worker support possibilities: context relevance, document relevance, action prediction and diversity. It is possible that providing recommendations for each event-block is too often. This should be optimized in future work. The reason that we choose to provide recommendations for each event block is that this represents the dynamic nature of the context well. In the next subsections we will describe an evaluation criterion with an evaluation metric for each of the support possibilities.

4.3 Context relevance

A first possible criterion in the evaluation of a context-aware recommendation system involves the evaluation of whether the suggested documents fit the user's current context. We aim to help the user focus on his activities, so suggestions that are related to a different context would possibly distract the user. In this evaluation measure, we define a correct context as a topical match between a suggested document and the current activities.

For this evaluation criterion we use the topic labels that are assigned to each document. These topic labels can be seen as a category of `context' and are equal to the topic-labels of the event-blocks. If the category of a suggested document matches the category of the current activities (e.g. the current event-block), we consider the suggestion to be a good one. We assess the quality of the recommendations using precision@10 (how many recommendations in the top 10 have the correct context).

4.4 Document Relevance

Although a topical match to the active context is interesting, it does not mean that a document that is suggested can be used by the knowledge worker. For example a knowledge worker producing a manual for some software will use different sources then when he is writing a report on the project for which the software was produced even though the context is the same. Therefore we consider the criterion of document relevance, which evaluates how relevant a suggested document is for the specific task the knowledge worker is working on.

Ordinary document relevance can be assessed by obtaining relevance judgements. However, for context-aware systems document relevance judgements need to be obtained within the context that the document was accessed. This means that we would need a document relevance assessment for each document in each context, and for each user separately. These relevance judgements are hard to obtain and are not available in the dataset.

An alternative to using manual relevance judgements is to look at the dwell time for each document. The advantage is that these are measured within context, so if a document is accessed within multiple contexts, multiple dwell times are measured. We investigated the appropriateness of dwell time in the dataset as criterion for relevance. If we use a threshold of 30 seconds (Guo and Agichtein, 2012), then only 44 documents in our data set would be estimated as relevant. This is only 1.3% of all documents in the dataset, which seems unrealistically low. One explanation comes from copy-paste behaviour. Some users tend to quickly copy some text from a viewed document to the document they are producing. This makes the dwell time for the viewed document low, even though the copy-behaviour suggests that the document is highly relevant. Also, when users quickly switch between the viewed document and the document they are producing, the individual dwell times are low.

Recent work has shown the limitations of using a (single) dwell-time threshold as relevance indicator and other evaluation metrics should be taken into consideration (Lehmann et al., 2013; Kim et al., 2014). In the dataset we use (described in Section 4.2) there was a strong focus on the production of texts. When we interpret document relevance as those documents that contain text that is used by the participants, we can use textual overlap between a suggested document and a produced document as an indicator for relevance of the document. The assumption is that the more relevant a document is, the more similar it will be to the produced document. This captures copy-paste behaviour that we observed in the data as well, since there will be a high similarity when complete sentences or paragraphs of one document occur in the other. Using this approach we can obtain personalized context-aware document relevance scores for each participant.

The limitation of this measure is that a produced document needs to be available in order to determine the relevance.

For this purpose we use the ROUGE-N measure by Lin (2004). This measure is originally intended for the evaluation of summaries or translations. It uses the number of overlapping n-grams between a source and a target document and is defined by:

$$score = \frac{2 * |source \cap target|}{|source| + |target|}$$

where we use the set of word 2-grams in the recommendation as source and the set of word 2-grams in the written document by the participant as target. In our interpretation a high ROUGE-score means that the document that is considered had a high contribution to the document that was produced by the user. In the original version of the measure, the score is normalized on the length of the user-produced document. However, as there can be a large difference between the length of the user-produced document and the candidate document, we normalize the score on the length of the sum of the documents. This length normalization is performed after stop-word removal [3].

Each produced document by a participant was tagged with the corresponding task context in which it was produced (the context labels of the event blocks). There were typically 6 produced documents per participant, one for each of the main tasks in the data collection. There were no produced documents for the tasks `Einstein' and `Information Overload'.

For each candidate recommendation, the ROUGE score is calculated between the candidate document and the produced document of the participant that was tagged with the label of the active context (i.e. the label of the event-block for which the recommendations are generated). In the case of web documents, html tags are removed before the ROUGE score is calculated. When there was no produced document available (i.e. `Einstein' and `Information Overload' or if a participant had not produced a document for the task), then the document relevance was automatically 0.0.

We assessed the validity of ROUGE-N as measure for document relevance on a randomly selected subset of 80 documents in their context. Two human assessors were shown two documents at a time: the produced document by the participant (the context), and the document to assess.

They were asked to provide a rating on a 5-point scale on how relevant the assessment document was for the creation of the produced document. There was a significant positive correlation between the ratings and ROUGE-N (Kendall's τ = 0.663, $p<0.001$), which means that a higher ROUGE-N score is associated with a higher human rating. Furthermore there was a substantial inter-annotator agreement on 20 overlapping items (weighted Cohen's κ = 0.68 (Cohen, 1968)). The positive correlation indicates that ROUGE-N can be used as measure for document relevance.

[3] Stopwords retrieved from http://snowball.tartarus.org/algorithms/english/stop.txt

4.5 Action Prediction

With the third evaluation criterion we evaluate the known item recommendation as an action prediction problem; which document will the user access next? If we can predict this document, and would present it to the user, this would save him the time to locate the document. We evaluate this by looking at the document the user accesses in the next event block. Since not all suggestions lists contain the document that will be accessed next, we consider success@1 and success@10: does the top-1 or top-10 list of recommendations contain the document that will be opened next?

4.6 Diversity

With the fourth evaluation criterion, we evaluate how original a document suggestion, or a list of suggestions is. This is in part contradictory to the relevancy criteria, since a diverse set of recommendations is more likely to contain distracting documents. However, we think that diversity is important in order to engage the user with the system. With a large enough document set, diversity should be possible without losing relevance.

We evaluate diversity by looking at two aspects: uniqueness of elements and variation between suggestion lists. Uniqueness is motivated as follows: if a recommender system offers more unique recommendations in one event block compared to the surrounding event blocks it is more original to the user then when it provides the same recommendations over and over again. For this aspect we consider how many unique items are recommended in all event blocks with the same context (a measure of catalog coverage (Ricci et al., 2011)) . This is measured with:

$$score = \frac{1}{C} \sum_{x \in C} \frac{unique_x}{total_x}$$

for a context $x \in C$, where $unique_x$ is the number of unique documents that occur as suggestion for a context, and $total_x$ is the total number of documents that have occured as suggestions for a context.

The second aspect is variation between suggestion lists. If subsequent suggestion lists are highly similar (e.g. the same suggestions in the same order), regardless of the actual activities of the user, the suggestions may not impact the user. Then the user will not consider the new suggestion list as original and he will not look at it. For this aspect we consider Rank Biased Overlap (RBO) as measure for rank correlation (Webber et al., 2010). RBO measures the similarity in ordering between two lists and is calculated using:

$$score = (1 - p) \sum_{d=1}^{n} p^{d-1} A_d$$

Where d is the position in the list, n is the size of the list and A_d is the proportion of the two lists that overlap at position d. The parameter $p = 0.9$ models the user's persistence (will a user look at the next item in the list). This gives more importance to the top ranked items than to the lower ranked items. This measure has the benefit that it is not hindered when there is no or little overlap between

the top 10 results (compared to other rank correlation measures such as Kendall). If there is no overlap than the RBO score is 0.

5 Method

In this section we describe three different approaches to context-aware information recommendation. Sections 5.1, 5.2 and 5.3 describe the three approaches and their implementation with the used dataset. Their effectiveness is evaluated and discussed in Section 6.

5.1 JITIR system

We implemented a just-in-time IR system as follows: For each user all 450 candidate recommendation documents were first indexed using the Indri Search Engine[4]. We used the Indri API to set up a query interface. For each event block in the data a query was constructed. This query consisted of the typed keys, window title and the text from the url or document that was active. All characters that are not alphanumeric, no hyphen or white space are removed from the query terms. As ranking model we used the default Indri Retrieval Model. The top 10 results, or less when there were less than 10 results, were considered for evaluation.

The JITIR system is hypothesized to perform well on document relevance, as it is has a focus on finding documents that contain terms that have been recorded in the current context as well.

5.2 Content-based Recommendation with contextual pre-filtering

We implemented a content-based recommendation system (CBR) with pre-filtering as means to incorporate context-awareness. This type of system is dependent on a (manual) categorization of the active context and the candidate documents in order to filter the candidate documents.

The dataset provides manually assigned context labels for each event block in the data. These labels correspond to the topics from the knowledge worker tasks(e.g. `Napoleon', `Healthy living'). Each document was assigned one or more context labels based on the labels of the event blocks in which the document was open. During run-time, the subset of documents with the same context as the event-block was selected. Then the items in the subset were ranked based on their cosine-similarity to the document that was open in the event-block. The features that were used were the normalized TFIDF scores on all terms in the documents. Documents that were more similar to the open document were assumed to be more relevant. When there was no document open in the event-block, the documents with the correct context were ordered at random.

We hypothesize that the active filtering of items with the wrong context has a positive effect on the performance on the context relevance criterion.

[4] http://www.lemurproject.org/indri/

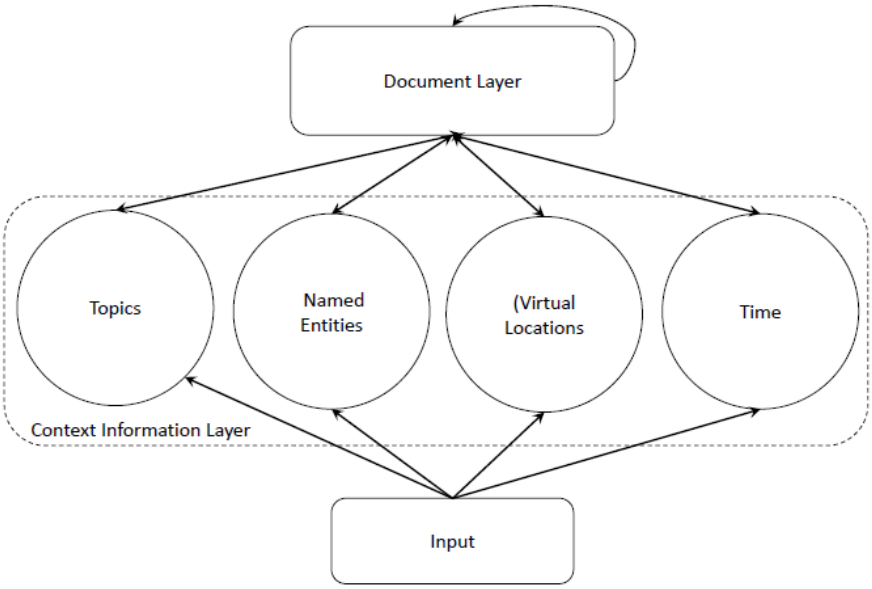


Fig. 1 The Contextual IA model (CIA). The figure shows the 3 layers. There are activating bidirectional links between the context information layer and document layer and unidirectional links from input to context. There are no links within layers, except for activating links between documents that are based on temporal closeness between opening documents

5.3 Context-aware Recommendation with Context Detection

The context-aware recommendation system with context modelling for context detection that we implemented is a novel method based on the interactive activation model by McClelland and Rumelhart (1981) and depicted in Figure 1. The added benefit of this method compared to CBR is that it does not depend on a manual source to determine what context is presently active. Compared to JITIR it has the benefit that it takes the history into account using decay.

In essence an advantage of the CIA network approach is that it could function as a memory extension for the user: The network stores explicit associations between information entities, similar as how the user would associate items. The idea of nodes, connections and spreading activation has relations to the working of the brain (Anderson and Bower, 1973). This could potentially benefit the recommendation, as it can use similar contextual cues for recommending items as a person would have used.

The network consists of three main layers:

- *the document layer:* this layer contains nodes for all 450 candidate recommendation documents This corresponds to the access history of approximately 25 hours (assuming on average 18 documents per hour)
- *the context information layer:* this layer contains nodes for the context information, divided into four categories of context information types: terms or topics, entities, locations and date/time elements
- *the event layer:* this layer is the input for the network. Here the sensed/recorded event-blocks enter and activate the network. In our dataset an event-block is a collection of events (key activity, mouse activity, window title, url) that was recorded within one tab or window of an application. This means that the event blocks are variable in their duration.

In the network the event layer activates the context information layer by observing which terms, entities, virtual locations and time information are present

Table 3 Connection strengths between the various node types. These are the weights on the activation ow from one node to another. They are based on the concept tf-idf term weighting.

From	To	Value or function	Motivation
Event-block	Date/Time	1.0	An event has one unique timestamp
	Entity	$\frac{\#entity_x \in event}{\#entities}$	Strength of activation of an entity should depend on how strong the entity is present in the event, proportional to the number of entities
	Location	1.0	An event has at most 1 location
	Topic	$\frac{\#topic_x \in event}{topic_{1..n}}$	Strength of activation of a topic should depend on how strong the topic is present in the event, proportional to the number of topics
Date/Time	Document	$\frac{1}{\#outlinks}$	Multiple documents can be accessed on the same date, or hour.
Entity	Document	$\frac{1}{\#outlinks}$	entities that occur in many documents should be less influential
Location	Document	1.0	
Topic	Document	$\frac{1}{\#outlinks}$	topics that occur in many documents should be less influential
Document	Date/Time	1.0	
	Entity	$\frac{\#entity_x \in document}{\#entities}$	Strength of activation of an entity should depend on how strong the entity is present in the document, proportional to the number of entities
	Location	1.0	A document only has one location
	Topic	$\frac{\#topic_x \in document}{topic_{1..n}}$	Strength of activation of a topic should depend on how strong the topic is present in the document, proportional to the number of topics
Document	Document	$\frac{1}{\#outlinks}$	

in the recorded event block. In its turn the activated context information nodes activate documents that are described by this context information; for example a term node `health' activates all the documents that contain the word `health'. Then the activated documents enhance the activity in the network by activating all their context nodes, for example the document that was just activated because it contained the word `health' now activates the word `well-being' as well because that word was also present in the activated document.

This spreading activation method serves as a sort of pseudo relevance feedback, mitigating the sparseness of the information in the event blocks. However, due to the sparseness of incoming information, there is a risk that too much irrelevant information is activated in the document layer. To prevent this `snowball effect' of sparseness, we implemented a TFIDF-like weighting for the connection weights: The connection weights from context information to documents are based on in-verse node frequency and the connection weights from document nodes to context information nodes are based on node frequency. As a result, observed context in-formation that occurs in many documents has less impact than information that occurs in only one document. And information that occurs frequently in a document has a bigger impact than information that only occurs once in the document. There are only positive (excitatory) connections in the network. A detailed motivation for the choice of connection weights can be found in table 3.

For each event-block the network is activated for 10 iterations. The difference in activation from one iteration to the next is defined using Grossberg's activation function:

$$\delta a = (max - a)e - (a - min)i - decay(a - rest)$$

where a is the current activation of a node, e is the excitatory input of the node, i is the inhibitory input and $min, max, rest$ and $decay$ are general parameters in the

model. The excitatory input pushes the activation to the maximum, while the inhibitory input drives it down to the minimum. The decay parameter gradually forces the activation back to its resting level when there is no evidence for the node and allows for cross-over of network activation from one event-block to the next. For pragmatic reasons, the network is not run until convergence, but only for 10 iterations. This is enough for sufficient activation in the network. The assumption is that the documents with the highest activations after those 10 iterations are the best candidates for suggestion.

In this paper we compare 2 variations of the CIA approach that vary in the method that is used to determine which topics or terms are representative for the context of interest. In the first approach, CIA-t, we use the top 1000 terms from the term extraction method described in Verberne et al. (2013) as representative terms. These 1000 terms are also extracted from events and documents.

In the second approach, CIA-lda, we use a latent dirichlet allocation model (LDA model) instead of term extraction to model the topics. LDA is often used for topic extraction. In this setting we have used the MALLET implementation of LDA (McCallum (2002)) and 50 topics are extracted. The initial LDA model is trained for 1500 cycles on a set of manually selected Wikipedia pages (e.g. the Wikipedia page `Napoleon' for the topic Napoleon), one for each of the tasks from the experiment. The same topics are also extracted from events and documents. For both CIA-t and CIA-lda we use the Stanford Entity Recognizer trained for English (Finkel et al., 2005) to determine which entities occur in event blocks or documents . The values of the parameters are the same as in the original IA network: : $min = 0.1, max = 1.0, rest = -0.1, decay = 0.1$

The CIA system in general is expected to perform well on diversity as it incorporates a form of query expansion, which allows for unexpected suggestions. This will be a trade-o with context relevance and document relevance, as more original suggestions will have a higher risk of being less relevant.

Another criterion on which CIA is expected to perform well is the prediction of which document a user is going to open. This is because CIA incorporates direct associations between documents, based on previous document access as well as document content.

Since the evaluation metric for document relevance is based on term overlap, we expect that CIA-t has an advantage over CIA-lda on the document relevance criterion as CIA-t also has a focus on terms rather than topics.

6 Results

For the discussion of the results, this section is divided into the four subsections that correspond to the four evaluation criteria described in Section 4. We compare the three methods described in the previous section to a baseline recommender

system that randomly selects 10 documents to suggest from the list of all 450 candidate documents. Documents that are open or have been opened before in the same session were excluded from the list of candidate documents. All significance values reported in this section are based on a paired samples t-test with a 95% confidence interval. The results are the macro averages over the event-blocks. Thus, first the average per participant is calculated. Then the average of these averages is reported to ensure that each participant has an equal effect on the average, regardless of the number of event-blocks in his session. The macro aver-aging method provides an estimate of the simulated system performance on each evaluation criterion averaged across 25 participants, using recorded standardized task guided -- but natural -- interaction data for approximately 3 hours (including short breaks).

The CIA-lda method uses an LDA model for topic recognition. Since LDA is non-deterministic, there could potentially be a difference in results between different initializations of the LDA model. Therefore, the reported results of CIA-lda are averaged over 5 runs, with 5 different LDA models. The differences between runs are not significant: $p = 1.000$.

6.1 Context Relevance

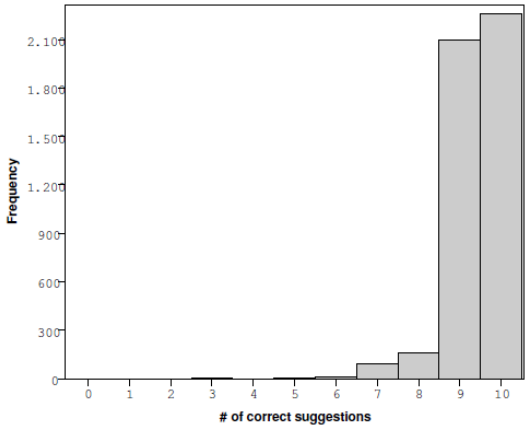
Table 4 Accuracy of the context of the suggestion.

Measure	CBR	JITIR	CIA-t	CIA-lda	Random
Precision@1	97.7%	59.1%	36.0%	44.2%	20.7%
Precision@10	94.1%	50.0%	39.7%	40.0%	19.6%

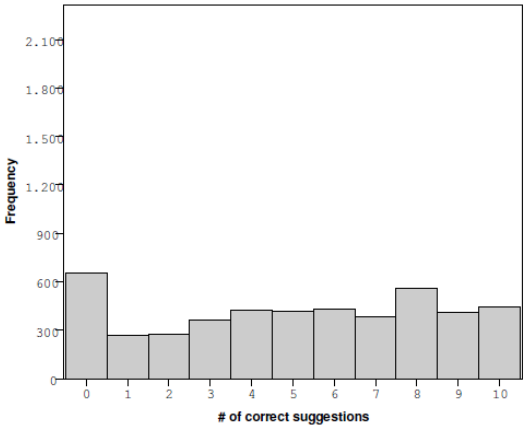
Table 4 shows the results for the match of the recommendation to the context. In addition, we present histograms for each recommendation method that show how often, how many of the 10 suggestions have the right context (Figure 2).

The table shows that the CBR approach is most effective in finding suggestions that match topically to the context (e.g. where the label of the document matches the label of the event-block). This is trivial as the CBR uses a hard filter on the context. Nevertheless, the histogram in Figure 2(a) that there are also event blocks for which CBR cannot provide 10 correct suggestions. In those cases, CBR does not have enough candidate documents remaining for the context after filtering the documents that have already been opened in the session. This happens in 51.3% of the event blocks.

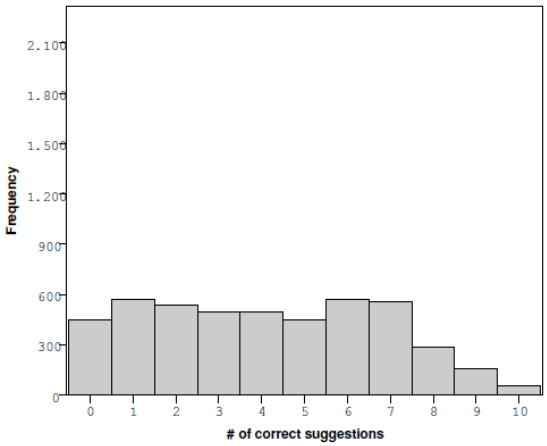
The JITIR system has a top document suggestion with the same topic as the active context in 59.1% of the cases. When the entire list of 10 suggestions is evaluated, 5 out of 10 suggestions have the right context on average. Both results are significantly lower than CBR ($p < 0.001$). Both the CIA-t network, and the CIA-lda networks have significantly lower success rates for its top recommendations (36% and 44.2% respectively, $p < 0.001$). CIA-t and CIA-lda suggest approximately 4 out of 10 suggestions with the right context, which is significantly lower



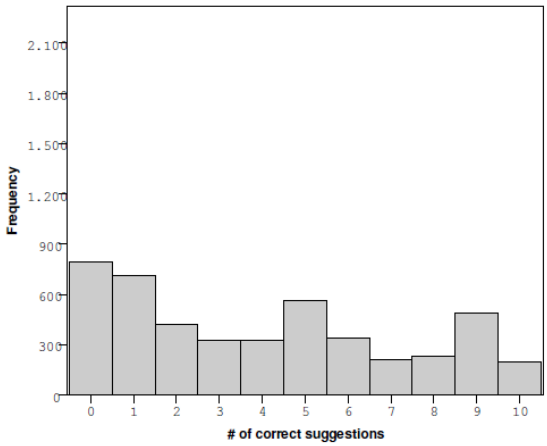
a) CBR



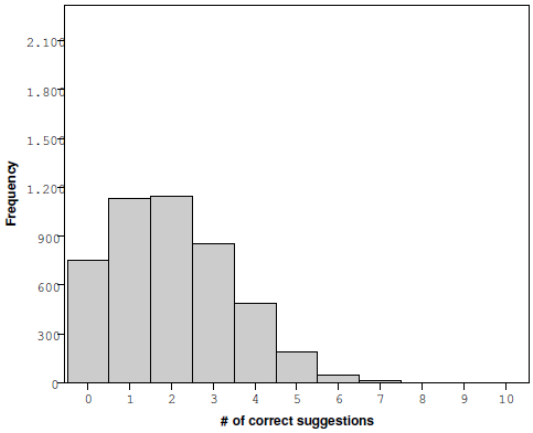
b) JITIR



c) CIA-t



d) CIA-lda



e) Random Recommendations

Fig 2 Histograms of the context relevance of suggestions

than CBR and JITIR ($p < 0.001$). The JITIR system, however, cannot suggest any documents in 3.9% of the event blocks, because of query-failure. CIA can always suggest the requested amount of documents, provided that there are sufficient candidate documents.

The random approach shows that on average 2 out of 10 suggestions will have the correct context when picked randomly, which is significantly lower than CBR, JITIR, CIA-t and CIA-lda ($p < 0.001$). The histograms in Figure 2 show that both CIA and JITIR show more uniform distributions over the number of correct suggestions, while CBR has a clear peak at 9 and 10 correct suggestions. The random system typically has 0, 1 or 2 correct suggestions within its suggestion list.

Since the CIA approach attempts to classify the context at the same time as it recommends documents, it is possible that there is a relation between the number of suggested documents with the right context and whether the context was accurately predicted. A one-way anova revealed that indeed the average number of suggestions with the correct context is significantly higher when the correct context was predicted ($p < 0.001$). For CIA-t 4.8 out of 10 suggestions had the correct context in the case of a correct prediction, while in case of a wrong prediction 2.8 suggestions were correct. For CIA-lda the difference was slightly smaller: 4.5 out of 10 for correct predictions, and 2.9 for wrong predictions.

6.2 Document Relevance

Table 5 Relevancy of the suggestion lists measured with ROUGE-N. (max) denotes the score for the best suggestion in the list, while (avg) denotes the average score for the entire list

Measure	CBR	JITIR	CIA-t	CIA-lda	Random
to written (max)	0.0149	0.0172	0.0117	0.0083	0.0053
to written (avg)	0.0031	0.0049	0.0023	0.0020	0.0005

Table 5 shows that regardless whether the complete suggestion list, or the best item in the list was considered, the recommendations by JITIR were most valuable (avg = 0.0049 and max = 0.0172). These values are significantly better than CBR, CIA-t, CIA-lda and random ($p<0.001$). A score of max = 0.0172 indicates that the textual overlap between the best candidate in the list and the produced document is 1.7% on average (over event blocks and participants). In this dataset, the maximum relevance that can be obtained for the best candidate document in a suggestion list for an event block is 0.6830. This is the ROUGE-score for a candidate--context--participant combination where the participant copied a large part of the candidate document in a particular task context. However, generally the scores are much lower: 84% of the candidate--context--participant combinations have a ROUGE score of 0%.

When the performance of the other systems is considered, CBR suggests more relevant documents (max and avg) than CIA-t, CIA-lda and random ($p<0.001$). Moreover CIA-t suggested more relevant documents than CIA-lda ($p<0.001$), which is what we expected, considering that CIA-t has a stronger focus on term overlap because of its term extraction method. This suggests that methods that have a strong focus on term matching such as JITIR have an a priori advantage on this metric.

Finally, both CIA-t and CIA-lda suggested more relevant documents than the random system ($p<0.001$). The random system has an especially low performance

if the entire list is considered (avg=0.0005), which is close to the average ROUGE score of 0.0007 for all candidate—context--participant combinations.

6.3 Action Prediction

Table 6 Predictive power of user's action.

Measure	CBR	JITIR	CIA-t	CIA-lda	Random
Success@1	0.0041	0.0053	0.0197	0.0253	0.0001
Success@10	0.0163	0.0148	0.0477	0.0483	0.0014

Table 6 shows that CIA-t and CIA-lda have better predictive power than JITIR and CBR. The success@10 measure reveals this; CIA-t and CIA-lda will predict the next document correctly in its list of suggestions in 4.8% of the cases (the difference between them is not significant: $p = 0.564$). JITIR only predicts the next document correctly in 1.5% ($p<0.001$). CBR predicts the next document with 1.6% predictive accuracy, which is comparable to JITIR ($p = 0.502$) and worse than CIA ($p<0.001$). When the top suggestion is considered (success@1) CIA-lda performs better than CIA-t ($p = 0.008$). Both CIA-t and CIA-lda are better than CBR and JITIR ($p<0.001$) and JITIR is better than CBR ($p = 0.023$). All systems are better than random ($p<0.001$).

Note that these predictive accuracies are rather low. This is a side effect of the requirement in the systems that they cannot recommend documents that are opened in the session already. Since the key log data includes frequent switches back and forth between documents, many of the recurrent openings of documents cannot be predicted. The theoretical maximum average predictive power is 42.6%, since 67.4% of document access events are of the type re-opening during the session.

Even though CIA, and specifically CIA-lda, performs the best on action prediction, CIA is potentially harmed by the manner of document selection. In the document selection, documents accessed by all participants are included in the list of candidate documents. However, for most of these documents, the simulated access data of the participant is not available. This means that the time nodes and document to document connections that CIA would normally create during online learning of interaction have not been created for the simulated experiment. These type of connections are especially relevant for predicting which document is going to be opened.

Table 7 shows the predictive power of the various methods, if the access pattern would have been available. This data is based on an experiment where only the documents accessed by a participant are included as candidate documents (on average 44 candidate documents). The table shows that the success rates of all methods improve because of the reduction in number of documents. CIA benefits the most: the success@10 of both CIA-t and CIA-lda increase to 12%, while the theoretical maximum predictive accuracy remains 42.6%. Interestingly in this case the difference between CIA-t and CIA-lda on Success@1 is not significant anymore ($p= 0.629$)

Table 7 Predictive power of user's action, personal document set.

Measure	CBR	JITIR	CIA-t	CIA-lda	Random
Success@1	0.0245	0.0170	0.0503	0.0498	0.0025
Success@10	0.0636	0.0570	0.1247	0.1238	0.0165

6.4 Diversity

Table 8 Variability in the suggestion list. Rank Biased Overlap (RBO) is measured with p = 0.9. RBO was measured for a suggestion list compared to all other lists in the session, as well as a suggestion list compared to the suggestion list of the next event. A low RBO value represents a larger diversity

Measure	CBR	JITIR	CIA-t	CIA-lda	Random
RBO – Session	0.468	0.195	0.245	0.331	0.059
RBO - Next Event	0.465	0.137	0.135	0.238	0.059
Unique Suggestions	12.6%	12.5%	14.2%	12.8%	17.9%

Table 8 shows that the suggestions by the random system have the highest variability (this means lowest RBO) in their orderings, which is expected given the current definition of diversity. CBR on the other hand shows a high RBO for both the session and the next event (47% commonality between lists). This can be explained by the filtering on context that CBR uses, which limits the choice in candidate documents per event-block. CIA-t, CIA-lda and JITIR score a bit in between in terms of variability. For session variability there is a 33% commonality (RBO) between recommendation lists for CIA-lda, 25% for CIA-t and 20% for JITIR, whereas the commonality from one event to the next is 24% for CIA-lda, and 14% for both CIA-t and JITIR.

In terms of unique suggestions, the random baseline has the highest number of unique documents in its suggestion lists, followed by CIA-t. The difference between the number of unique suggestions for CIA-lda, CBR and JITIR is minimal. Overall, CIA-t scores slightly better on this criterion than CIA-lda, JITIR and CBR because of more unique suggestions in combination with a low RBO score between events.

7 Discussion

We have presented four evaluation criteria that are relevant for the evaluation of knowledge worker support in the task of information re-finding. In Section 7.1 we start with a discussion of the evaluation measures to answer the question “How should we evaluate a context-aware information recommendation system in light of the goal to support knowledge workers in re-finding information?" (RQ1)

In Section 7.2 we continue with a discussion of three context aware recommendation approaches and their performance on the four evaluation criteria. This answers the question what the benefits and downsides are for the various approaches for recommending documents with the purpose of helping the knowledge worker. (RQ2)

We conclude with a discussion on the limitations of this work and some suggestions for future work in Section 7.3.

7.1 Evaluation Criteria and Metrics

The evaluation criteria that were described in this paper cover several aspects of knowledge worker support. Some of these evaluation criteria may be related. For example, if a document is relevant for the user, it is not likely that this document will distract the user (i.e. does not match the active context of the user). Therefore, we measured the correlations between the metrics that were used to assess the four evaluation criteria.

A two-tailed Pearson correlation test reveals that context relevance is moderately positively correlated with average document relevance ($\rho = 0.445$, $p < 0.001$). This means that indeed a document that does not fit the current activities is not likely to be relevant.

The other measures have negligible correlations. Action prediction is negligibly correlated with context relevance ($\rho = 0.040$,$p<0.001$) and document relevance ($\rho = 0.062$,$p<0.001$). Diversity as measured with rank biased overlap (RBO) is negligibly uncorrelated with document relevance ($\rho = 0.008$, $p < 0.001$) and action prediction ($\rho = 0.018$, $p=0.187$), but weakly positively correlated with context relevance ($\rho = 0.122$, $p<0.001$).

These correlations suggests that the document relevancy measure might be redundant. We should, however, in the future look at a situation where there are multiple writing tasks with similar topics, to fully understand the document relevancy measure. Nevertheless, since some of the context-aware recommendation methods are focused on using context categories, while others use a more elaborate context, it seems reasonable to evaluate both tasks separately. Otherwise there might be a bias towards the type of context-aware approach already.

Moreover, there are still aspects of the ROUGE-metric for document relevancy that need to be considered. In this paper we have used stop-word removal, length normalisation and the removal of html tags as preprocessing steps for the ROUGE-metric. An aspect that we have not considered is the selection of text that needs to be considered for the metric. For example, wikipedia indicates the part of the page that is being watched with a suburl (i.e. https: //en.wikipedia.org/wiki/Napoleon_Bonaparte#Early_career). However, the text that is used for the calculation of document relevance is based on the entire web-page, since the webcrawler extracts the entire page, not just the part that is being watched. In general the crawled webpages are a source of noise. Sometimes the actual text cannot be extracted, for instance when the page is in Flash.

Another point for discussion is the definition of the diversity measure. At this point diversity is measured independent of relevance. However, recommending diverse but irrelevant documents is not beneficial for the knowledge worker. This

shows that it is important to consider the various evaluation criteria in combination. By measuring them in isolation, an incomplete picture about the performance of a system is sketched, which becomes apparent when we consider the performance of the random system on the diversity measure.

Overall, when we consider the knowledge worker and his situation as a whole as described in the scenario we prefer a method that scores well on all evaluation criteria. After all, a system that can prevent distractions really well (context relevance) is not useful when it only suggests the same documents over and over again (diversity). A system that can predict which documents will be opened is not useful when these documents will distract the user.

7.2 Context-aware recommendation approaches

When we consider the performance of the various context-aware recommendation approaches on the four evaluation criteria, we can conclude that the preferred recommendation method should depend on the task at hand. In a complex scenario such as the knowledge worker scenario, the preferred recommendation method can vary even in a single day of work, to optimally support the variety of activities that the knowledge worker is involved in. Therefore, it is important to continue to work towards a context-aware recommendation approach that scores well on all tasks and is not dependent on context assignments.

If the goal of the system is to prevent distractions for the knowledge worker (context relevance), the content-based recommender system with contextual pre-filtering (CBR) shows the best results. This supports the hypothesis that CBR is good at preventing distractions because it actively filters documents with the wrong context. This result is trivial, and illustrates why it is important to consider multiple evaluation criteria. Also note, that although a context match implies that the document is no distraction, a document with the wrong context does not need to be a distraction if it provides relevant information for the task (e.g. a document that is tagged with `Stress' could also be relevant for the active context `Healthy Living').

If the goal of the system is to suggest documents that are likely to contain relevant information that the knowledge worker can use, then JITIR is the best choice, both when the complete suggestion list is considered as well as when only the best item in the list is considered. For this criterion systems which suggest documents that textually overlap with the current context have a benefit.

If the goal of the system is to predict which documents a knowledge worker will open, then CIA is the best choice, especially when the document access patterns are available (which is the default case, since CIA has been designed to take advantage of interaction patterns). This supports our hypothesis that CIA has an advantage in action prediction because there are direct associations between documents based on the time-of-opening in the CIA approach. Moreover CIA provides top results in document relevancy.

If the goal of the system is to provide a high diversity in results (regardless of relevance), then the random system should be used. This is a result that could be expected given the current definition of diversity. CIA shows promise in terms of diversity as well, especially when term extraction is used (CIA-t). CIA-t suggests more unique documents than CIA-lda, JITIR and CBR. Of course, the fact that

the diversity measure does not take relevance into account is a limitation of the measure. Overall CIA, JITIR and CBR are preferable over the random system as they will recommend more relevant documents by design.

Regardless of their performance on the evaluation criteria, each recommendation method has advantages and disadvantages. CBR has the advantage that it is a simple and robust method. However, CBR is sensitive to a cold-start problem that occurs for every new context that is introduced.

If there are no or few documents that are tagged with the active context, than CBR cannot provide a sufficient amount of recommendations. Because of the hard filter, CBR cannot use documents that are tagged with a different but strongly related context, even though these might be good suggestions. Furthermore, CBR depends on a manual source to determine which active context is currently active, which requires more user effort. This is especially the case in the knowledge worker scenario, where the context is highly dynamic.

The advantage of JITIR is that it does not depend on an external source for context determination. The use of context as query is simple and effective, and there is no need for context categorization. The downside is that sometimes this query fails, so that no recommendations can be provided. This occurs in 3.9% of the event blocks.

In essence an advantage of the CIA network approach is that it could function as a memory extension for the user: The network stores explicit associations between information entities, similar as how the user would associate items. With this mechanism it is a step towards the design principles formulated by Elsweiler et al. (2007) to improve personal information management systems. The disadvantage of CIA, is that its recommendation lists have a lower context accuracy. However, the flexibility of the method can be used to improve performance on certain criteria such as document relevance, for instance by using term extraction instead of topic modelling.

When we consider the complete knowledge worker situation as described in the scenario, we judge CIA as the most promising approach of the three. CIA is good at predicting which document the user will access next and provides a diverse set of recommendations. Although its recommendation list might contain documents that do not strictly match the current context, overall it seems to contain at least one good suggestion for most event blocks.

7.3 Limitations and Future Work

One aspect of evaluation that is lacking is the real-time performance and scalability of the approaches. This is an aspect that is important when a system is put to practice, especially for context-aware systems. A system is not likely to be useful to the knowledge worker when the suggestions are not provided in time. Since our dataset contained not enough data to pose problems for scalability and did not contain data over multiple days, we have not considered these dimensions in this paper.

A further limitation of the research presented in this paper is that there was only one dataset that we could use. Its characteristics may explain some of the generally low performances on document relevancy. The document data in the set was not filtered for noise and contained data in at least two languages.

Additionally, the dataset contained no actual relevance judgements, causing us to divert to derivative measures. For a proper evaluation of the methods and evaluation metrics it, we should look at the performance on a second dataset.

In future research it is important to investigate what users value most in context-aware recommendation systems. How often should the system recommend documents, and how many documents should be in the suggestion list? This means most likely, that recommendations are not independent of each other and should thus also not be evaluated as such (Zhai et al., 2003).Another important aspect is the further exploration of the evaluation metrics that we have used. Are these the optimal ones, or are there alternatives that have a stronger relation to the user's preferences? Although we have presented an alternative to dwell-time and document relevance judgements in the form of ROUGE-N, its characteristics need to be explored further to see the potential of the measure as alternative measure for document relevance. Moreover, the diversity metrics should be adapted to take the relevance of the suggestions into account in order to make the criterion less trivial.

Furthermore we propose to consider a task-dependent cost-based metric in the future to determine which recommendation strategy to use at a certain time. The cost should be dependent on the characteristics of the task the knowledge worker is executing. This would allow the design of a hybrid context aware recommendation system that can optimally support the knowledge worker in various circumstances. For example it could stimulate diversity when the knowledge worker is exploring a new topic, while focusing on context relevance when the knowledge worker needs to finish a task.

8 Conclusion

In this paper we have described the evaluation of context-aware recommendation for information re-finding with the purpose of supporting knowledge workers. The scenario of the knowledge worker is different from typical context-aware recommendation scenario's as the context is more dynamic and there is larger negative impact of irrelevant recommendations. In this paper we have presented and used a dataset that facilitates research to this kind of complex recommendation scenario's. We focus on four evaluation criteria that are relevant for knowledge worker support context relevance, predicting document relevancy, predicting user actions and diversity of the recommendation lists.

We have evaluated three different approaches to context-aware document recommendation in a realistic knowledge worker setting where the context is given by the interaction of users with their regular office PC. One approach to context-aware document recommendation is a content-based recommender with contextual pre-filtering (CBR), one is a just-in-time information retrieval system (JITIR) and one is a novel method that is capable of detecting the active context simultaneously to providing context-aware document suggestions (CIA).

The conclusion of what context-aware document recommendation method per-forms best highly depends on the evaluation criterion that is considered. Overall, each method performed well for at least one evaluation criterion. CBR was best at context relevance, JITIR was best at providing a recommendations that are likely to contain text that the knowledge worker will use and CIA was best at predicting

which document the user will open. The random baseline was best at providing diversity in its suggestions.

Overall we believe that the CIA approach is most promising for context-aware information recommendation in a re-finding setting as it performed best in terms of action prediction, while providing diverse results as well. Moreover, CIA is not dependent on a manual source for detection of the active context. Nevertheless, there is room for improvement when it comes to document and context relevance. The flexibility of the system provides ample opportunities to investigate these aspects. Finally, we conclude that the multi-faceted evaluation approach has added value in complex task based evaluations.

9 Acknowledgements

This publication was supported by the Dutch national program COMMIT (project P7 SWELL).

References

Anderson, J. R. and Bower, G. H. (1973). *Human associative memory*. VH Winston & Sons.

Bawden, D. and Robinson, L. (2009). The dark side of information: overload, anxiety and other paradoxes and pathologies. *Journal of information science*, 35(2):180--191.

Blanc-Brude, T. and Scapin, D. L. (2007). What do people recall about their documents?: implications for desktop search tools. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 102--111. ACM.

Budzik, J. and Hammond, K. J. (2000). User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th international conference on intelligent user interfaces*, pages 44--51. ACM.

Cai, F., Liang, S., and de Rijke, M. (2014). Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 835--838, New York, NY, USA. ACM.

Chen, Y. and Jones, G. J. F. (2014). Are episodic context features helpful for refinding tasks?: Lessons learnt from a case study with lifelogs. In *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14, pages 76--85. ACM.

Cohen, J. (1968). Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213--220.

Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., and Robbins, D. (2003). Stuff i've seen: a system for personal information retrieval and re-use. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72--79. ACM.

Dumais, S., Cutrell, E., Sarin, R., and Horvitz, E. (2004). Implicit queries (iq) for contextualized search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 594--594. ACM.

Elsweiler, D., Ruthven, I., and Jones, C. (2007). Towards memory supporting personal information management tools. *Journal of the American Society for Information Science and Technology*, 58(7):924--946.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363--370. Association for Computational Linguistics.

Gao, A. and Bridge, D. (2010). Using shallow natural language processing in a just-in-time information retrieval assistant for bloggers. In *Artificial Intelligence and Cognitive Science*, pages 103--113. Springer.

Gomez-Perez, J., Grobelnik, M., Ruiz, C., Tilly, M., and Warren, P. (2009). Using task context to achieve effective information delivery. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, pages 3:1--3:6. ACM.

Guo, Q. and Agichtein, E. (2012). Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web*, pages 569--578. ACM.

Henzinger, M., Chang, B.-W., Milch, B., and Brin, S. (2005). Query-free news search. *World Wide Web*, 8(2):101--126.

Ingwersen, P. and Järvelin, K. (2006). *The turn: Integration of information seeking and retrieval in context*, volume 18. Springer Science & Business Media.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79--86. ACM.

Kelly, D. (2009). Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(12):1--224.

Kelly, L., Chen, Y., Fuller, M., and Jones, G. J. (2008). A study of remembered context for information access from personal digital archives. In *Proceedings of the second international symposium on Information interaction in context*, pages 44--50. ACM.

Kim, Y., Hassan, A., White, R. W., and Zitouni, I. (2014). Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 193--202. ACM.

Lai, C.-H., Liu, D.-R., and Lin, C.-S. (2013). Novel personal and group-based trust models in collaborative filtering for document recommendation. *Information Sciences*, 239:31--49.

Lakiotaki, K., Matsatsinis, N. F., and Tsoukias, A. (2011). Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64--76.

Lehmann, J., Lalmas, M., Dupret, G., and Baeza-Yates, R. (2013). Online multitasking and user engagement. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 519-- 528. ACM.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74--81.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

McClelland, J. L. and Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. an account of basic findings. *Psychological review*, 88(5):375.

Melguizo, M. C. P., Bajo, T., and Gracia Castillo, O. (2010). A proactive recommendation system for writing in the internet age. *Journal of Writing Research*, 2(1).

Oku, K., Nakajima, S., Miyazaki, J., and Uemura, S. (2006). Context-aware svm for context-dependent information recommendation. In *Proceedings of the 7th International Conference on Mobile Data Management*, MDM '06, pages 109--, Washington, DC, USA. IEEE Computer Society.

Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme, L. (2011). Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 635--644. ACM.

Rhodes, B. J. (1997). The wearable remembrance agent: A system for augmented memory. *Personal Technologies*, 1(4):218--224.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

Sappelli, M., Verberne, S., and Kraaij, W. (2013). Recommending personalized touristic sights using google places. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval,* SIGIR '13, pages 781--784, New York, NY, USA. ACM.

Sappelli, M., Verberne, S., S.J., K., and Kraaij, W. (2014). Collecting a dataset of information behaviour in context. In *Proceedings of the 4th Workshop on Context-awareness in Retrieval and Recommendation*.

Verberne, S., Sappelli, M., and Kraaij, W. (2013). Term extraction for user pro-ling: evaluation by the user. In *Proceedings of the 21th International Conference on User Modeling, Adaptation and Personalization*.

Wakeling, S., Clough, P., and Sen, B. (2014). Investigating the potential impact of non-personalized recommendations in the opac: Amazon vs. worldcat.org. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 96-- 105. ACM.

Warren, P. (2013). Personal information management: The case for an evolutionary approach. *Interacting with Computers*, pages 208--237.

Webber, W., Mo at, A., and Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems* (TOIS), 28(4):20.

Weng, S.-S. and Chang, H.-L. (2008). Using ontology network analysis for research document recommendation. *Expert Systems with Applications*, 34(3):1857--1869.

Zhai, C. X., Cohen, W. W., and Lafferty, J. (2003). Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10--17. ACM.