XXXX

Adapting the interactive activation model for context recognition and identification

Maya Sappelli, TNO and Radboud University Nijmegen
Suzan Verberne, Radboud University Nijmegen
Wessel Kraaij, TNO and Radboud University Nijmegen

In this paper, we propose and implement a new model for *context recognition and identification*. Our work is motivated by the importance of 'working in context' for knowledge workers to stay focused and productive.

A computer application that can identify the current context in which the knowledge worker is working can (among other things) provide the worker with contextual support, e.g. by suggesting relevant information sources, or give an overview of how he spent his time during the day.

We present a descriptive model for the context of a knowledge worker. This model describes the contextual elements in the work environment of the knowledge worker and how these elements relate to each other. This model is operationalized in an algorithm, the *contextual interactive activation model* (CIA), which is based on the interactive activation model by Rumelhart and McClelland. It consists of a layered connected network through which activation flows. We have tested CIA in a context identification setting. In this case the data that we use as input is low-level computer interaction logging data.

We found that topical information and entities were the most relevant types of information for context identification. Overall the proposed CIA-model is more effective than traditional supervised methods in identifying the active context from sparse input data, with less labelled training data.

1. INTRODUCTION

In our project SWELL[1] we aim to support knowledge workers in their daily life. One aspect is their working life. With the increasing amount of information they have to handle, knowledge workers can get overwhelmed easily: a phenomenon referred to as 'information overload' [Bawden and Robinson 2009], and filtering irrelevant information or 'working in context' is deemed beneficial [Gomez-Perez et al. 2009; Warren 2013]. Additionally, with the arrival of smart phones and "any place, any time information" (e.g. the wish and opportunity to access information at any place at any time), proper work-life balance is at risk.

This creates two use cases for supporting knowledge workers. In the first use case ('working in context') we aim to support knowledge workers by filtering information based on their current activities (providing contextual support). Ardissono and Bosio [2012] have found that task-based and context-based filtering reduce the user's current workload. Thus, by recommending and highlighting information that is relevant to the context, while blocking information that is out-of-context, we help the user to stay focused on his current task.

In the second use case ('user-context awareness') we aim to make users aware of their activities and work-life balance, by showing them a record of their activities. A concrete example is by means of 'hour tracking'. Many companies ask their employees to define how much time they spend on each project during a week for cost definition purposes. By providing the user with an automatic overview of his day or week, the employee can save time on this task.

Both use cases, that are closely related to life logging [Gurrin et al. 2014], require us to keep track of what the knowledge worker is doing during the day. That is, we aim to identify the user's *context*.

Unfortunately, context is a vague concept. Many researchers [McCarthy 1993; Akman and Surav 1996; Dervin 1997; Penco 1999; Brézillon and Pomerol 2001; Dey et al. 2001; Dourish 2004] have tried to define the concept, but it seems difficult to get a

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

good grip on it. There is no single answer to what the ‘context’ in ‘working in context’ is Brézillon [1999a,b] nor how it can be recognized automatically. Therefore, we analyse related literature on context and context modelling in section 2.1. We are not only interested in what context should look like for our application, but also how the user’s activities can be mapped to meaningful contexts. That is why we present an overview of existing approaches to recognizing context automatically in section 2.2.

We present our own definition and descriptive model of context in section 3. These provide the starting point for a computational model of context. The main contribution of this paper is a novel approach and implementation for context *recognition* and *identification* which is described in section 4. Compared to existing approaches, this method aims to keep the effort to use the system as low as possible. This means that little or no labelled data is required to initialize the method, which ensures that we do not add to the load of the knowledge worker.

Our research questions are:

RQ1: How can we model the context of a user and what are the requirements of this model?
RQ2: How can we implement the model for context identification in a way that requires a minimal amount of labelled data for training?
RQ3: What information is required for successful context identification?
RQ4: How effective is our model in identifying the user’s context?

2. BACKGROUND AND RELATED WORK

In this section we present an overview of literature on context in personal information management and context recognition and identification approaches.

2.1. Context in Personal Information Management

Context is a concept that is often used, but rarely defined. Since this is a possible source for miscommunication, we provide some background on context and how we interpret context in the remainder of this paper. We describe literature from the field of personal information management (a sub field of Information Retrieval and Information Science), as this area is most relevant for the support of knowledge workers.

In the research area of personal information management, Gomez-Perez et al. [2009] define context as “a set of information objects that are frequently accessed concurrently or within a very short time-span”. Additionally, information objects that are similar in terms of content may belong to the same context as well. They stress that for “working in context” to be helpful, relations between information sources in the same context need to be meaningful to the knowledge worker and therefore they leave the actual definition of context (e.g. which groupings of objects are relevant) to the user. Sauermann [2009] adopt this view in their semantic desktop approach to personal information management. Their model is a personal information model (PIMO) built by the users themselves that is used across applications and domains.

In contrast, several researchers have adopted a view on context that is not dependent on the personal interpretation of the user [Ermolayev et al. 2010; Whiting and Jose 2011; Devaurs et al. 2012]: The ontological context model for knowledge workers by Ermolayev et al. [2010] is based on a pragmatic selection of things that are related to the entity on which the context is about. These include processes (for example development of a controller) and objects such as persons, resources, tools etc. Whiting and Jose [2011] share this view. They attempt to provide contextualized recommendations of previously accessed information sources and summarize the contextual elements they use for that purpose. These are fixed and measured independent from the user’s beliefs.

Devaurs et al. [2012] also seem to agree with this view. They present an ontological model for context, but do not take into account the user. Ingwersen and Järvelin [2005] describe their nested model of context stratification (p281) in which they centralize context around a core and see multiple dimensions of context around this core. In a sense these dimensions are all nested containers. Additionally, the core can be either an object or a person, suggesting the possibility for both a subjective and an objective view on context. Finally, Schmidt et al. [2011] distinguish the internal (human intention) from the external context (not related to human intention) and the unperceived elements. They use explicit task models to assess the intention of a user. The models can be created by the user, or generated automatically.

In our approach, we will centre context around a user, allowing a subjective interpretation of what this context should entail. We will sense context in an objective setting, independent of the user. However, we will take the user's actions into account in determining the importance of the sensed elements, to maintain a subjective focus. In order to do so we will define some contextual elements of which we think are relevant for the application, similar to previous approaches. Since we determine the active contextual elements based on sensed events, the context detection becomes highly dynamic.

2.2. Context Recognition and Identification

In the previous section we summarized some literature on the concept of context from the field of Personal Information Management. Now we review the literature into the process of automatic context recognition as it is an important element in context-aware personal information management systems. In the presented literature, context recognition is essentially the mapping of one or multiple events (such as the user's active windows and typed keys) to a label that can be interpreted by the user as a meaningful activity. We actually see this as context *identification* rather than *recognition*. From the literature on context we have learned that context usually entails a collection of many elements, thus context recognition should be the recognition of these elements, and not the process of summarizing these elements in a communicable label. In the remainder of this paper we will use context identification when we refer to the representation of context by a single label, while we will use context recognition when we describe context by all its elements.

In any case, the context identification methods presented in literature vary in the interpretation of what type of identification is interesting. The different types of context identification methods we look at are: topic-based (section 2.2.1), process-based (section 2.2.2) and memory-based (section 2.2.3). We present our own approach in section 4.

2.2.1. Topic-based context identification. The methods for topic-based context identification focus on generalizing events to topics. For example by identifying some computer activities as related to "trip to Rome" vs. "trip to Paris".

There are several approaches. A first group of studies essentially sees context identification as a *categorization* problem. These approaches are similar to document categorization as they typically monitor the terms in the documents, document sequence, or window title and map them to one of the context categories. The classification algorithm varies from network-based (WordSieve by Bauer and Leake [2001]), graph-based (SeeTrieve by Gyllstrom and Soules [2008]), Bayesian classifiers (IRIS by Cheyer et al. [2005] and TaskTracer by Stumpf et al. [2005]) to SVM (Task Predictor by Shen et al. [2006]).

Secondly, there are approaches based on clustering where the process is mainly about finding clusters of related documents or windows and evaluating these on labelled data. In the Swish-system by Oliver et al. [2006] windows are clustered using latent semantic indexing, in ACTIVE (Warren et al. [2010]; Štajner et al. [2010]) the authors use a weighted sum of cosine similarity for term overlap, social-network overlap and temporal proximity of documents and document access. In ACTIVE the document or information object is central; context identification is simplified to recording the cluster-tag that is given to the active information object at cluster-time.

In a third approach by Maus et al. [2011], context identification is primarily a manual process, done in their system Contask, which is integrated in the 'Nepomuk Semantic Desktop' (Groza et al. [2007]). Users define tasks and can associate information objects with these tasks. The users themselves are responsible for maintaining the appropriate active context thread, however Contask does provide a service where context switches are automatically detected with the purpose to propose the user to initiate a context switch.

The reported accuracies and precision-recall values are difficult to compare as each author evaluated his algorithm on small and private datasets. There is no publicly available dataset to compare results because of privacy concerns related to the data.

In these works, the main source of information is document content. In our work we propose to use keystrokes, mouse clicks and window information as well as the content of documents and other information objects as input variables for context recognition and identification.

2.2.2. Process-based context identification. In this section we describe literature on process-based context identification methods. These focus on identifying a context by generalizing the process that is involved in the example. For example by identifying some activities as "planning a trip" vs. "claim expenses". Compared to the topic-based approaches the classes vary in the process that is involved, rather than the subject of the activity as was the case in "trip to Rome" vs. "trip to Paris", which would both be classified as "planning a trip" in the process-based approach.

For this type of context identification the approaches are similar to the topic-based approaches (2.2.1). Granitzer et al. [2009] use a traditional classification approach in which they compare the performance of Naive Bayes, linear Support Vector Machines (SVM) and k-Nearest Neighbour classifiers (k-NN). Naive Bayes performed best for estimating the five tasks that the authors had defined, while k-NN with $k = 1$ performed best in estimating labels defined by the participants themselves.

Devaurs et al. [2012]; Rath et al. [2010] also compare Naive Bayes and k-NN classifiers as well as J48 decision trees and linear SVM. In addition to features from the information objects in the data, keyboard strokes, mouse events and other interaction features are used in the classifier. These features are managed in their ontology-based user interaction context model, UICO. The best classification results were obtained with J48 decision tree and Naive Bayes classifiers.

A clustering method is described by Brdiczka [2010]. Their task reconstruction system uses a spectral clustering algorithm to find task clusters based on the temporal switch history.

Furthermore, Armentano and Amandi [2012] used Variable Order Markov models with an exponential moving average to predict the user's goals from unix commands.

Koldijk et al. [2012] use a key logger to monitor a knowledge worker's activity with the purpose to track which tasks the user is performing. They investigate in a user study which task labels the knowledge workers intuitively use. These tasks include: read or write e-mail, write report, program, analyse data and search for information. Additionally, they investigated whether these tasks can be recognized automatically from the low level log events (such as mouse or key activity or the active application) using automated classifiers (SVM, Naive Bayes,etc.). They found that with relatively little data, i.e. a few hours, reasonable classification accuracy of 60–70% , depending on the user, could be obtained. However, there were many individual differences and there was no single classifier type that performed consistently over users.

In the SWELL project we use the work by Koldijk et al. [2012] to provide feedback to the user on the activity level, but for our identification of context we are more interested in a topic-based identification. The combination of feedback on activity level and on context/topic-level gives the best insight on how the user has spent his day, which is our goal in the 'user-context awareness' use case.

2.2.3. Memory-based context recognition. Some authors interpret context recognition merely as a memory process, and only use temporal information to recognize contexts. They do not identify a context as label, but as a combination of tasks that were active at the same time: they memorize which windows were previously open together with the current window.

An example is the study by Abela et al. [2010], they propose a task-based user model that acts as a knowledge workers' mental model of a task, consisting of all computer resources related to that task. These should be used to resume a task-state after it has been suspended. The authors indicate the problem that different documents opened in the same application may belong to different tasks, complicating the method to be used for making a task snapshot.

Additionally, Omata et al. [2010] propose a project-restarting system where files associated with a main file are automatically reopened. Associations between windows containing files and the importance of the window are automatically predicted. Features they use are window depth, visible representation ratio, and screen occupancy ratio.

Kersten and Murphy [2012] describe a task-focused desktop in which they present users with lists of documents associated with the tasks. The list is trimmed based on the frequency and recency with which a user interacts with the associated documents to determine whether it is still interesting for that task. They describe a longitudinal case study in which some university colleagues work with their system. The users manually start and stop tasks, during an active task all accessed documents are automatically associated with the task. The authors find that their users tend to revisit tasks mostly the same day, suggesting that there is no need for auto-trimming.

These studies suggest that the memory-based approach is useful when we aim to support the user's work flow, but it is not usable to present the user with an overview of his day (use case 'user-context awareness'). Therefore, we focus on topic-based context identification instead.

3. THE KNOWLEDGE WORKER'S CONTEXT

In the previous section we have summarized how context can be interpreted in a personal information management system and what type of context identification approaches can be distinguished. As our application goal is to make a user aware of how he spends his day, it is important that we identify context in a manner that is meaningful to the knowledge worker. A topic-based identification of context will be most suitable for that goal. In order to develop a method for topic-based context identification, it is necessary to model the meaningful aspects in the knowledge worker's context. In this section we will present a descriptive model of the knowledge worker's context. We simplify the knowledge worker's environment to a collection of resources with which a knowledge worker interacts in order to accomplish his work tasks. We do not aim to provide a holistic and complete model of the worker's context, but aim to provide a bigger picture on elements that can influence the work of a knowledge worker. This includes some elements that need further exploration and are not yet used in the computational model for context recognition and observation in Section 4

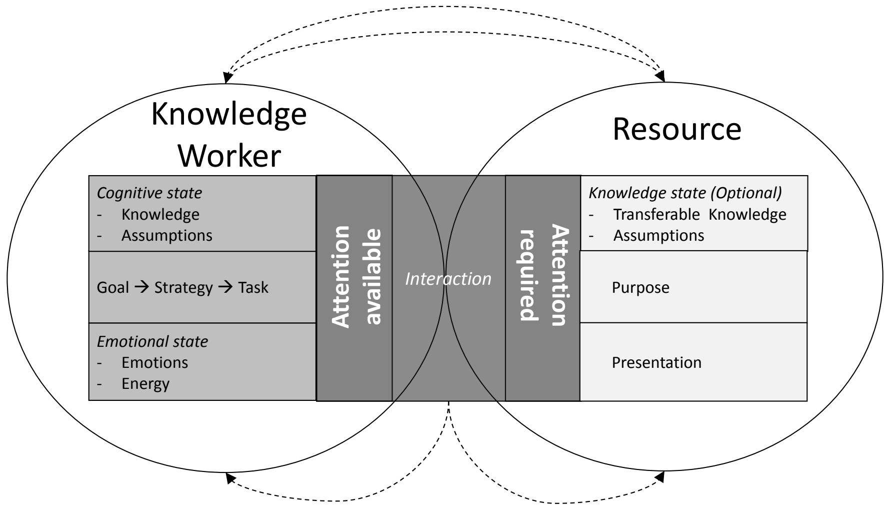

Fig. 1. Descriptive model of context for a knowledge worker. Arrows depict that one element influences the other.

(e.g. attention, emotion). However, these additional elements do provide part of the motivation for the flexible approach taken in Section 4 and can be seen as possible future extensions as described in 4.2.

When we consider the context of a knowledge worker, the user (the knowledge worker) is the central point of reasoning and we are interested in those elements that influence the user. However, it is still unclear which elements are important and how they interact with each other. In Figure 1 we present our view on the knowledge worker and his context in a descriptive model. We consider a knowledge work environment. Typical to this environment is that it includes a knowledge worker, who is a person whose main job is to produce and distribute knowledge, and one or more resources. Examples of resources are: another person, a device such as a computer, a (printed) paper, but also the lighting in the location where the knowledge worker is. The knowledge worker interacts with these resources to achieve his or her goals. Goals are achieved by formulating strategies, which consists of one or more tasks. We assume that a knowledge worker is a person who is characterized by:

(i) a task that the knowledge worker wants to execute,
(ii) a cognitive state: consisting of (1) general knowledge that the knowledge worker has obtained by education or in previous experiences and (2) assumptions about the knowledge work environment in general and the resources in it in particular,
(iii) an emotional state: the emotions and energy a knowledge worker has, and
(iv) a limited amount of available attention or focus at each moment in time in order to acquire knowledge.

We assume that a resource is characterized by:

(i) a purpose for which it was created,
(ii) a knowledge state: consisting of (1) knowledge that can be transferred and (2) in the case of personalized systems, assumptions about the knowledge worker,
(iii) presentation (in what condition is the resource), and

(iv) a required amount of attention or focus at each moment in order to transfer knowledge.

The knowledge worker K can choose to interact with some resource R. This interaction can change the state of the knowledge worker as well as the resource. It is rationally guided by the task K wants to execute, and constrained by the purpose of the resource. Furthermore, the emotions of K can influence the interaction. This is especially the case when the resource is another person.

Typically, the reason for the interaction is to transfer knowledge or information from the resource to the knowledge worker. However, the interaction is influenced by the amount of attention that the knowledge worker has available. Some contextual elements will not be perceived consciously, because K is not focusing on it. Also a resource R can have a conscious or unconscious effect on the knowledge worker, even when he or she does not explicitly interact with R, for example a flickering light that distracts the knowledge worker. Overall, the conscious and unconscious effects of resources, and the attention that K has available determines which knowledge and information is actually transferred.

We assume that a knowledge worker engages in interaction with the resource that most likely has the best positive influence on the current task. In the automatic recognition and identification of context, we focus on the observable information that is transferred from a resource to a knowledge worker when they interact. This means that we do not take into account emotions and attention in the remainder of this paper.

4. CONTEXT RECOGNITION AND IDENTIFICATION USING AN INTERACTIVE ACTIVATION APPROACH

The descriptive model in Figure 1 described the elements that play a role in the context of a knowledge worker. In practice, the way the contextual elements influence each other is complex. To evaluate the model in a more straightforward task, we identify what a user is working on. The method that we present, however, is designed to be able to also take into account more complex tasks in the knowledge worker context, and a more diverse range of contextual elements than which we evaluate in this paper.

For now, we describe a method to recognize and identify context. That is, we extract meaningful contextual information from the interactions with the computer (context recognition), and we attach a tag to it that the knowledge worker can interpret as one of the tasks he is working on (context identification). In the evaluation presented in Section 5 this task tag is the project name where the current activities belong to. We continue with a description of the contextual information that we use, after which we describe the novel context recognition and identification method.

4.1. Contextual Interactive Activation model (CIA)

In previous work (see Section 2.2), the information that has been used to recognize context can be categorized in the following dimensions: time, terms or topics, social information and location. In Figure 2 we visualize the literature that has been described in Section 2.2 in terms of the types of information that they have used. In contrast to previous literature, the SWELL project aims to integrate all four dimensions of contextual information. Especially when working with multiple sources of data it is important to realize that different types of information sources have different characteristics that need to be dealt with appropriately. For example, in e-mail the sender and receiver are very important for categorizing the message, while for documents the content is more important [Sappelli et al. 2012, 2014a]. Considering that both email and documents are important parts of a knowledge worker's activities, it is important

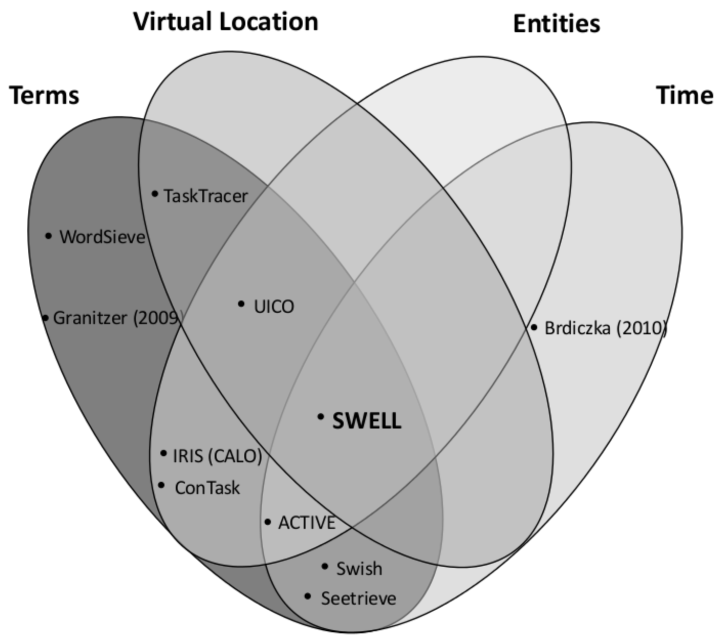

Fig. 2. Visualization of literature and which information each project uses to link documents or events. The information types are: terms (the terms or topics that occur), (virtual) location (e.g. same directory), entities (e.g. names, brands, cities etc.), or time (a temporal relation, e.g. accessed the same day)

to be able to use both topics as well as entities such as person names as inputs for context recognition.

The difficulty in context identification is how to combine the various dimensions in an effective manner. In the method that we describe we have chosen a cognitively plausible approach that associates contextual elements to each other without the need to explicitly define the relations between them. The human brain is constantly making associations between observations [Anderson and Bower 1973], and the intuition is that modelling these associations is the key to understanding how an individual would interpret his context. For example:

> The project 'SWELL' could be described by the terms 'stress', and 'knowledge worker' and the time period '2012'. If at some point in time the term 'burnout' is observed, we will most likely ascribe the logged activity to the project 'SWELL'. Although there is no direct association between 'SWELL' and 'burnout', we can find an indirect association between 'stress' and 'burnout', leading us to a correct classification of the observed activities into the project 'SWELL'

Another motivation is that these associations could give insight in the behaviour of knowledge workers in terms of context switches. For example:

> A person is reading about 'Turing' in relation to the turing test. But, through recent associations with the movie 'The imitation game', the knowledge worker is distracted, switches his context and reads up on the latest news about 'Benedict Cumberbatch'.

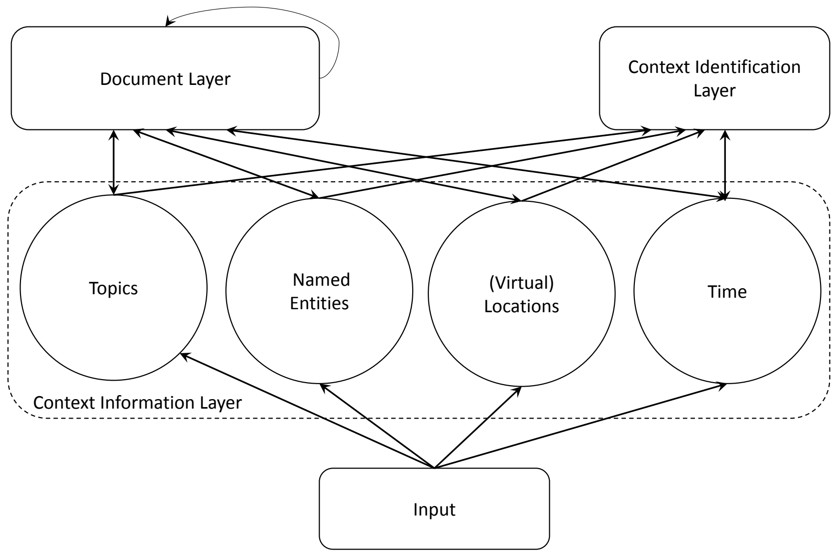

Fig. 3. The Contextual IA model (CIA) consisting of an input layer that activates the network (for example event blocks from a key logger), a context information layer (the information types from from Figure 2), a document layer, and a context identification layer as target output layer.

The intuition of the importance of associations has inspired us to adapt a well-known cognitive model for word recognition; the interactive activation and competition model (IA model) by McClelland and Rumelhart [1981]. The IA model is a feed forward model that assumes that letter features stimulate relevant letters, letters stimulate relevant words and finally words stimulate relevant letters again. Within each level there is competition; each feature inhibits other features, each letter inhibits the other letters etc. A rest activation and decay function complete the model. The model has been successfully applied as a cognitive model for bilingualism [Dijkstra et al. 1998] as well.

4.1.1. Construction of the initial network. We have adapted the IA model to use it for context recognition and identification (see Figure 3). Hereafter it will be referred to as the Contextual Interactive Activiation-model (CIA). It is constructed as follows: First, we define three layers in the model:

— the document layer: this layer contains all information objects that a user writes or reads and includes web-documents and emails. These can be complete documents, but could also represent paragraphs, especially when a document is very long.
— the context information layer: this layer contains the context information, divided into the four categories of context information types (terms or topics, entities, (virtual) locations and date/time elements) in Figure 2. In the model (Figure 1) this is the cognitive state: the knowledge of the user. This information can be obtained using automatic methods such as a named entity recognizer or a topic modeling method.
— the event layer: this layer is the input for the network. Here, recorded events from a key-logger, collected in event-blocks, enter and activate the network. In the model (Figure 1) this is the transferable knowledge of the resource. In the case of computer activity one possible instantiation of an event-block can be a collection of events (key activity, mouse activity, window title, URL) that was recorded between opening a tab

Table I. Connection strengths between the various node types. These are the weights on the activation flow from one node to another. They are based on the well-known Information Retrieval concept tf-idf term weighting. Other choices for connection strengths and interpretations are possible.

From	To	Value or function	Interpretation and Motivation
Event-block	Date/Time	1.0	An event has one unique starting time stamp, which can be decomposed in several date/time elements such as day of the week.
	Entity	$\frac{\#entity_x \in event}{\#entities}$	Strength of activation of an entity should be dependent on how strong the entity is present in the event, proportional to the number of entities. This can be determined using an entity extractor
	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 be dependent on how strong the topic is present in the event, proportional to the number of topics. This can be determined using topic modeling such as LDA or term extraction
Date/Time	Document	$\frac{1}{\#outlinks}$	Multiple documents can be accessed on the same date, or in the same hour.
Entity	Document	$\frac{1}{\#outlinks}$	IDF type measure; entities that occur in many documents should be less influential
Location	Document	$\frac{1}{\#outlinks}$	Multiple documents can be stored at the same location, for example a file folder or a webdomain. Location that contain many documents should be less influential.
Topic	Document	$\frac{1}{\#outlinks}$	IDF type measure; topics that occur in many documents should be less influential
Document	Date/Time	1.0	Date/Time that a document is first observed in an event. A copy of a document is considered a different instantiation, and is represented by a separate node.
	Entity	$\frac{\#entity_x \in document}{\#entities}$	Strength of activation of an entity should be dependent on how strong the entity is present in the document, proportional to the number of entities. This can be determined using an entity extractor
	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 be dependent on how strong the topic is present in the document, proportional to the number of topics. This can be determined using topic modeling such as LDA or term extraction
Document	Document	$\frac{1}{\#outlinks}$	Each document has a connection to the document that is accessed after it. If there is a strong temporal relation between documents, there will be fewer unique documents to which it is linked, and the related document will receive a strong activation.

or window of an application and switching to another tab or application or closing it. Only user-initiated events can open or close an event block, not system initiated events such as pop-ups. Event blocks can vary in their duration, for example editing a document, switching to the browser and returning to the document would result in 3 very short blocks. On the other hand, writing some text using resources other than a computer could result in an event block that spans multiple hours.

Each of these layers contains nodes, and each node contains connections to nodes in another layer. Each node is a nominal version of the variable. Time nodes include nodes for each year, for each month in the year (January–December), for each day in the week (Monday–Sunday), for each day in the month (1-31), for each hour in the day (0-24) and for each quarter in the hour (0,15,30,45). Each of the locations, entities, topics or terms has a single node. Since entity recognition and topic recognition can be probabilistic in nature depending on the method of choice, probabilistic properties can be enforced using the connections. For example, the probability that a topic is observed in an event determines the connection weight between that event and the topic. Similarly the probability that a topic is observed in a document determines the weight of the connection from the document to the topic. These probabilities can for example be approximated using bayesian inference.

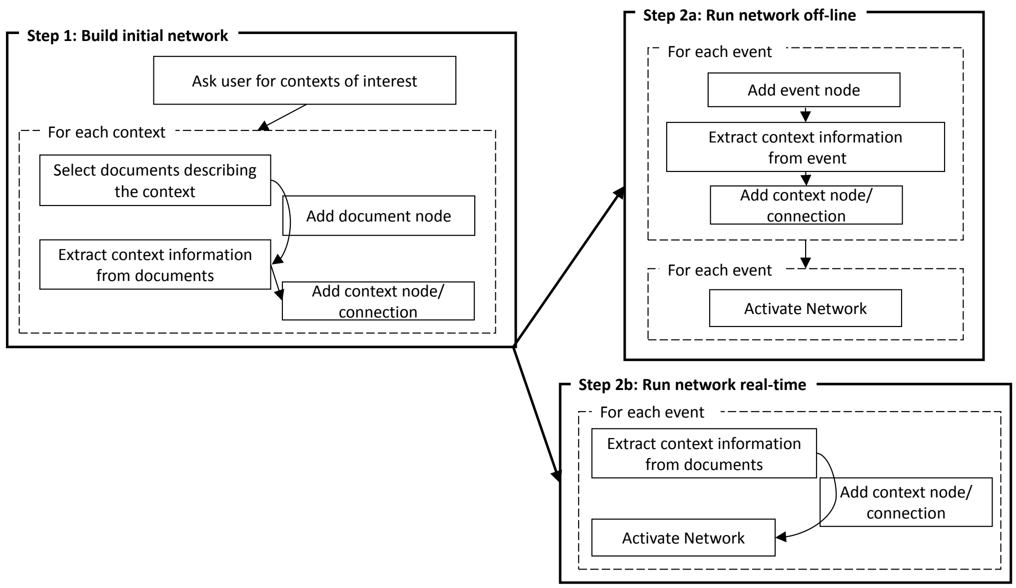

Fig. 4. Work flow of building and running the CIA model

An initial network is built top-down (step 1 in Figure 4). The documents on the computer of the user are collected and for each of them a node is created on the document layer. For each document the location attribute is recorded, and the topics and entities are extracted using topic and entity recognizers. These context information elements are represented by nodes on the context-level. Bi-directional connections are made between each of the extracted context nodes and the corresponding document node. The strength of the connections of document to context differs from the strength of context to documents. An overview of the connection weights and the motivation for the various weights is given in Table I. Presently we only use excitatory connections (stimulation connections) in our network and no inhibitory connections in order to limit the complexity of the model. Another reason is that in contrast to word recognition, multiple contexts can be validly active at the same time, so there is no need for competition between the contexts. There are no within-level connections at the context level. This is partly to keep the model simple and efficient, and partly because these connections are already there indirectly through the document level.There are within-level connections on the document level, which will be clarified later on.

4.1.2. Running the network. Essentially we have an initial model now that describes the associations an individual may have made based on information that is already on their computer. In this paper we focus on context identification to provide the user with an overview of his day ('user context awareness'), a task which is executed only once a day (step 2a in Figure 4). For this purpose we can enhance the model bottom-up with each event block (coming from a key logger) that is observed. Incoming event blocks are temporarily added to the event layer. From these event blocks context information is extracted to create context nodes and connections, similar as in the documents, but now based on the event information. There is always a date/time-stamp for the event. If there is sufficient content information, originating from window titles, typed keys and caption information, entities or topics can also be identified. If there is a reference to a document in the event block, this document is added to the document layer. Connections are made from the date and time elements of the event block to the document. Since temporal proximity has been used successfully as a feature for context recognition [Warren et al. 2010; Štajner et al. 2010], connections are made from the document in the current event block to the document in the previous event block. This is a within-level connection between two documents based on their temporal proximity. A risk with these connections is that the relation is not valid, as is the case when a user switches contexts. In our implementation, we have a weight based on the number of unique connections. If there is a strong temporal relation between two documents, both documents will have fewer connections with other documents than documents that do not have such a strong temporal connection. This means that the weight for valid connections will be higher than for the invalid connections. Another option could be to make the weights dependent on frequency of occurrence.

In contrast to adding the events to the event layer in the building process, we could also complete the network simultaneously while running it to provide real-time support (step 2b in Figure 4). This is necessary for the 'working in context' scenario where the user is supported during his work with context-aware functionality such as notification filtering or finding relevant documents.

Running the network entails that the event nodes (input nodes) are activated and that the activation is spread through the network using an activation function. The activation procedure has 3 steps:

— First the event block is activated in the event layer. The activation of the node is set to the maximum.
— Then the connected context nodes are activated using Grossberg's activation function, which runs for several iterations. The difference in activation from one iteration to the next is defined as follows:

$$\delta a = (max - a)e - (a - min)i - decay(a - rest) \quad (1)$$

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 input consists of $sum(w_ij * a_i)$, where w_ij is connection weight between node i and node j and a_i is the activation of node i. If $(w_ij * a_i) >= 0$ it is used as excitatory input, and $(w_ij * a_i) < 0$ for inhibitory input). The excitatory moves the activation towards the maximum, while the inhibitory input moves it towards the minimum. The decay parameter causes the activation to return 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.

— In the next iteration of the activation function these context nodes stimulate their connected nodes and this continues for several iterations such that all levels in the network get properly activated.

4.1.3. Learning associations between context and tag. The steps up to now allow us to activate the network, making it possible to *recognize* the active context in terms of an activation pattern over context information elements. We cannot, however, *identify* context yet. For that purpose we need some additional nodes; context identification nodes. These nodes represent the context identification tags described earlier. An example of suitable labels are project names, which we evaluate in Section 5.2.

In order to *identify* the active context, the model first needs to learn what the contextual elements are that are associated with a certain context identifier. These associations reveal which connections to make between context nodes and context identification nodes. The identification nodes have no outgoing connections. The activation level of an identification node signals which context was most likely active during the event block.

There are three possible approaches to determine which connections between context nodes and identification nodes need to be made and how strong they should be related. The first is a manual process where the user would be asked to describe each context identification tag with a couple of keywords or entities that are related to it. These terms are then the first context nodes in the network. A downside to this approach is that it is possible that the data, whether it be event blocks or documents, might not contain the exact descriptions of the user. The connected nodes might be very sparse. Additionally it is difficult to properly weigh the connections.

The second approach is a supervised one, where the network would be presented with a subset of event blocks that are labelled with their context identification tags. Connections can be made between the elements in the events (topic, time, location and entity) and the context identification tag with which the event block is labelled. This will result in more connections than in the manual approach, but the downside is that the event blocks from the key logger need to be labelled first.

The final approach is an alternative method and is related to question 2: "How can we implement the model for context identification in a way that requires as little labelled data as possible?". In this approach transfer learning is used. One method of transfer learning is a method where labelled items from a source domain are used to train a classifier in a (different) target domain [Arnold et al. 2007; Bahadori et al. 2011]. In the knowledge worker case the system makes use of documents on the user's file system as source domain, to be able to classify event blocks; the target domain. For that purpose, each project folder name on a user's computer can be used as a context identification tag. The documents in that folder can serve as the training data for the connections that need to be created between the contextual elements and the context identification tags. Thus, contextual elements are extracted from the documents in the project folder and connections are made accordingly. With this method, no labelled data, other than a couple of organized documents, is needed and the connections can be weighted according to their strength of occurrence in the documents. Using these knowledge worker's native structures in order to reduce modeling effort has been applied successfully by [Maus et al. 2005] as well. However, we take the approach one step further where we use it as a type of transfer learning where document categorization is used as a source for initializing a network for the purpose of the categorization of events into contexts [Arnold et al. 2007; Bahadori et al. 2011]. In essence the approach is of the type feature-representation transfer [Pan and Yang 2010]. The context layer in the network can be seen as a feature representation that represents both the source domain (documents) and the target domain (events) and reduces the difference between the two. We have applied and evaluated this method successfully in the domain of e-mail categorization [Sappelli et al. 2014a]. In the remainder of this paper we focus on using the network in a transfer learning setting.

For clarity we want to add that the learning aspect in our network, albeit unsupervised, supervised or by transfer learning, only entails learning which connections should be made between context level and context identification level, and not which weights are optimal.

4.2. Relation of CIA to the descriptive model

In Section 3 we provided a descriptive model for the context of a knowledge worker. In this section we will elaborate on the relation between the descriptive model and the computation model (CIA).

In our current implementation, CIA only uses the observable information in the interaction of a knowledge worker with a resource. This observed information is limited by interactional elements (events) and part of the transferable knowledge of the knowledge state of the resource (e.g. documents). In the long run, the network can be seen as an approximation of the knowledge in the cognitive state of the knowledge worker. This allows for reasoning about which information is new, and which is known, an important requirement for supporting a knowledge worker more optimally.

The reason for taking a network based activity approach for context identification and detection is that it allows for incorporation of the other elements from the descriptive model. For example, if a knowledge worker is low in energy or if the presentation of the resource is noisy, overall activity in the network could be suppressed. Furthermore, emotion nodes could be added to the network in order to represent feelings about certain topics or entities. If an emotion is observed during an event, the topics and entities associated with that emotion are more likely to be active, so the emotion can serve as a bias towards those topics and entities. Finally, attention can be taken into account by biasing or enhancing the pieces of information that are in focus. For all these enhancements, more research is required in order to understand how the element (emotion, energy, attention) should influence the activity in the network, and how we can observe it.

5. IMPLEMENTATION AND EVALUATION

In this section we will evaluate how well we can identify the context of the user using the proposed model. For the proposed 'user-context awareness' use case in Section 1 (giving feedback on how the user spent his time) it would suffice to evaluate classification power on hourly time frames. However, for the 'working in context' use case, where information needs to be filtered directly, context identification on small time frames may be needed, even though in that case the information in the context layer may also be used directly rather than to make an identification step first. Since the focus of this paper is on context identification and it is easier to go from small time frames to larger time frames, we evaluate the output of the network on a per event-block basis [Sappelli et al. 2014b], where the average duration of an event block is in the range of seconds.

5.1. Data

For the evaluation we use a dataset of event-blocks originating from human-computer interaction data representative for a knowledge worker's activities[2] which is labelled according to the activity that was carried out during the event block (Koldijk et al.

[2]Note that although the dataset was collected in an artificial work setting, it has several unique properties that make it representative for knowledge worker tasks: multiple tasks (creating presentations, writing

[2014]; Sappelli et al. [2014b]). This dataset is publicly available[3] The blocks were collected during a controlled lab-experiment where 25 students were writing reports and preparing presentations on 6 subjects (e.g. a road trip in the USA, Napoleon) while they were being monitored by various sensors. During the experiment two additional subjects were introduced, resulting in a total of 8 context identification tags, that we aim to recognize. In a real office settings these tasks would be the various projects that a user is working on. We only use the sensor-data from the installed key logger (uLog v3.2.5) and the file history (coming from IEhistory). The individual events from the key-logger and file history are aggregated in event blocks. Keylogging events that occur within the same application and window title are considered to belong to the same event block. The duration of the block varies from 1 second to 8 minutes. The e-mail messages in the inbox of the participants are not used as documents in this experiment. Typed e-mail messages, however, can be reconstructed from the typing events, and are part of the data that was used for the experiment.

Labelling. The data was labelled using Amazon Mechanical Turk. The annotators were presented with a mimicked version of the original desktop view of the event. Additionally a field with the typed keys was presented. The annotators were asked to choose one of 8 tags, corresponding to the subjects, and an additional 'unidentifiable' tag. They were also asked how certain they were of their answer on a 5 point scale, with 5 being completely certain and 1 being not certain at all.

The dataset consists of 9416 labelled event blocks, with an average of 377 event blocks per participant. The distribution of the labels, excluding unidentifiable labels, is quite skewed as can be seen in Figure 5. The labels 'Einstein' and 'Information Overload' have less event blocks, since these were not main tasks. The labels 'Perth' and 'Roadtrip' occur relatively often, most likely because these tasks required more searching and planning, and with that a higher variety in sources.

5.2. Implementation Details

For the extraction of topics from the documents and event blocks we use a latent dirichlet allocation model (LDA model), which 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. In a real office setting, these documents could be project description documents. Document inference (i.e. determining the topic for a new unseen document) is also based on sampling for 1500 cycles. The input for inference on an event consists of text from window titles, typed keys and captions.

For the entity extraction, the Stanford entity extractor trained on English is used. This trained model provides good results for the entity classes Person, Organization and Location, which are the ones we use in the network (Finkel et al. [2005]). Again inference is done on either document content or event content (text from window titles, typed keys and captions)

The date and time nodes on the context layer in the network consist of separate nodes for day of the week, day of the month, month, year, hour and minutes rounded to 00, 15, 30 and 45. The location nodes in the network are the file folder in the case of files on the computer and the domain name in case of web-documents.

reports, answering e-mail, searching for information and images), on multiple topics. The dataset, however, does not fully cover the potential real life setting for a knowledge worker

[3]http://dx.doi.org/10.17026/dans-x55-69zp

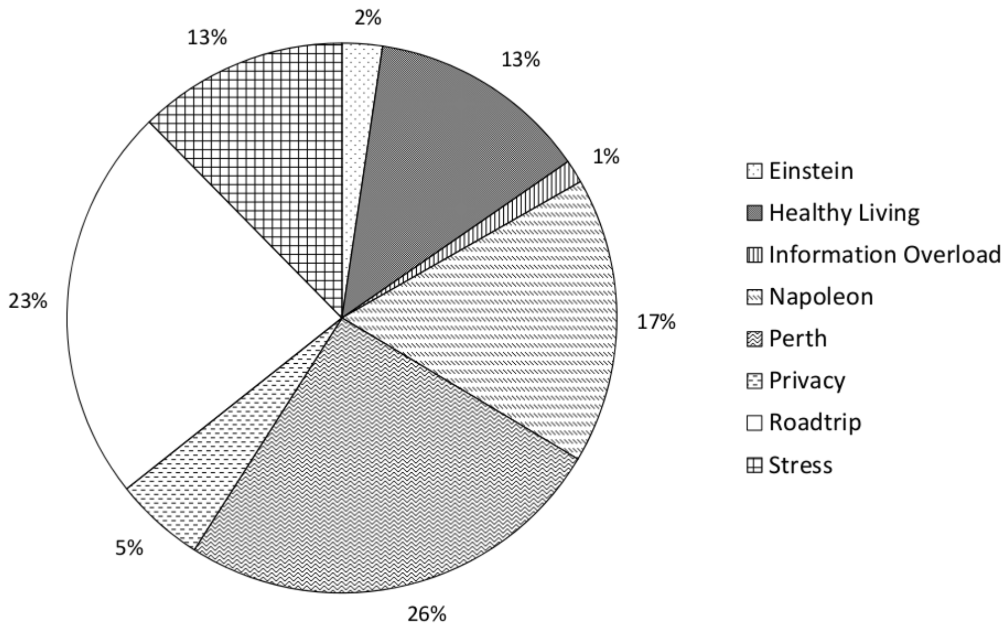

Fig. 5. The distribution of the identifiable labels in the data.

Normally the network would be run until the activation converges. However, this reduces the effect of the activation of previous event blocks (see Section 5.3). Therefore, and for efficiency we run each event block for 10 epochs (iterations). Considering that it takes 3 iterations to activate all the levels in the network, 10 iterations ensures that each level in the network is activated, that the recurrence in the network is activated, and that there is still influence of previous event blocks. Moreover it keeps the running time low. The number of activation epochs is a setting in the network that can be managed by the user. Efficiency is important for our application scenarios ('working in context' and 'user context awareness') as well, but in some settings we may want to distinguish between short – little impact – event blocks and longer event blocks, which are probably more important. In that case we could run each block for a number of epochs corresponding to the duration of the event block times 10. This means that an event-block with duration 1 second, is activated for 10 epochs, so that there is enough activation in each of the levels, but an event-block with a longer duration is run for 100 epochs and thereby has much more impact on the overall activation in the network.

The identification of a context is based on the node with the highest increase in activation for an event block, compared to the node's resting level of activation. This is necessary because the network does not necessarily converge within 10 epochs. By looking at the increase in activation rather than the highest absolute value of activation, the network focuses on the evidence in the event block. This ensures that nodes that are decaying are not preferred over nodes that have an increase in activity, even though the absolute activity of the decaying node may be higher than that of the increasing node. By using the increase relative to the node's own resting level we prevent that nodes which activity levels increase slightly because they are already highly activated are unjustly ignored. The resting level of nodes may vary due to the number and strength of incoming connections. Nodes that have many incoming connections are

Table II. Parameter settings used during evaluation

Parameter	Definition	Value
α	Strength of excitation	0.1
γ	Strength of inhibition	0.1
Min	Minimal value of activation	-0.2
Max	Maximal value of activation	1.0
$Rest$	Resting-level of activation	-0.1
$Decay$	Strength of decay	0.1

more likely to always receive a little bit of activation, preventing them from returning to the global resting level.

These settings have been tweaked based on the data of participant 2, whose data has also been used to optimize the parameters of the various algorithms (See Section 5.2.1).

5.2.1. Parameter Optimization. There are six parameters in the original IA model. Additionally, the LDA model we use has some additional parameters: the number of topics and the number of iterations. We used data of one of the participants (person 2) as development data, and used classification accuracy as our optimization measure. We first optimized the LDA parameters using the default IA parameters, resulting in an LDA setting with 50 topics and 1500 iterations. Then, using those LDA settings, we optimized the IA parameters with a hill-climbing approach starting from the default parameters. We found no set of settings that was significantly better than the default, so our final parameter set is the same as the default parameters from IA as presented in Table II.

5.3. Understanding the CIA approach

Before we dive into the performance of the network and compare it to baselines in the next section, we first want to show what is happening in the network at run time, and why we think this is useful for the problem at hand.

The main issue in our context classification problem is that the data that we can observe, namely the event blocks with key-logging information, is very sparse and noisy. Since we focus on window titles and typed keys there is not much data that can be used. A window title only contains a couple of words, and these are not necessarily related to the content of the window. Typed keys may include more words, that however could be expressed in the wrong language, or contain typing errors and corrections, resulting in incomplete or erroneous data. An example event block is presented in Table III. The network approach allows the expansion of the sparse observed data with information that is associated with the observed input. For example, a single recognized entity such as 'Australia' in event block 42 is likely to occur in some document about Australia. By activating the Autralia related documents, other entities such as 'Perth' and 'Rottnest Island' are activated, and activation of Australia-related topics is enhanced. This increases the likelihood that the correct label 'Perth' is assigned.

Not only is the data that we observe sparse, the data is represented by different types of features (i.e. the nodes) such as topics, entities, location and time information. It is important that the number of features for a type does not influence the result. For example, in the LDA model we have 50 features, while for the entity model we might have many more. The entity features should not outweigh the topic features, simply because there are more of them. In the network, features that are not activated have little impact on the overall activation. Thus during activation it does not matter that there are more entity features than topic features, since most entity features will not be observed and not activated.

Table III. Example Event Block, backspaces, key combinations and mouse activity are represented as (additional) spaces

id	42
time	20120919T133339573
app	WINWORD
window title	
typed keys	Australia is a cp ountry that a country knkw with a number of a c0 country know n for its con untless natural wonders va and a one among thr mos e amoin ng the top tpurist destinatioop n ns that the tourust visiti. s wa o scemo nic beauty. It Austrai l West Australia o Perh the th isth ca Australia consists of many beaitif utiful cities
(web-) domain	
(mouseover-) caption	Close Document1 - Microsoft Word Document1 - Microsoft Word

An important aspect in activating the network is the decay parameter. This parameter ensures that past information is not immediately forgotten. For context classification this is useful, because typically there is a dependency between one event block and the next, when the knowledge worker is working on the same task. The effect of decay is especially clear when the network is run for few iterations. The iterations help to smooth the boundaries between event blocks (Figure 6), because history-information is taken into account. When the network is run for 100 iterations the boundaries between event blocks are more clear (Figure 7) because in that case the focus is on the evidence from the current event block instead of the history. Both figures show that the Healthy Living and Stress labels are in competition with each other.

5.4. Results

In this section we analyse the model in terms of its classification performance of the event blocks. We start with a comparison of the proposed network, CIA, to existing approaches k-NN and Naive Bayes. Then we analyse the effect on performance of each of the context information types. We continue with a comparison of the network to one that is built on the fly and that can do real-time context identification. We end with the analysis of the influence of language on the performance of the model, since non-English knowledge workers often use a mixture of languages.

Table IV. Comparison of the CIA model to k-NN and Naive Bayes baselines

Method	Avg Accuracy
a. CIA using LDA	**64.85%**
b. CIA using Term Extraction	61.56%
c. k-NN with $k = 1$	55.83%
d. k-NN using LDA with $k = 1$	59.75%
e. Naive Bayes	48.32%
f. Naive Bayes using LDA	60.49%
g. Majority Baseline	25.30%

5.4.1. Accuracy of the CIA-model. In Table IV we present the accuracy of our CIA model with LDA topic recognition (a). These results are the average over the 25 participants over 25 runs per participant. The need to compare multiple runs per participant stems from the random nature of the LDA model (We will elaborate on this in Section 6). We

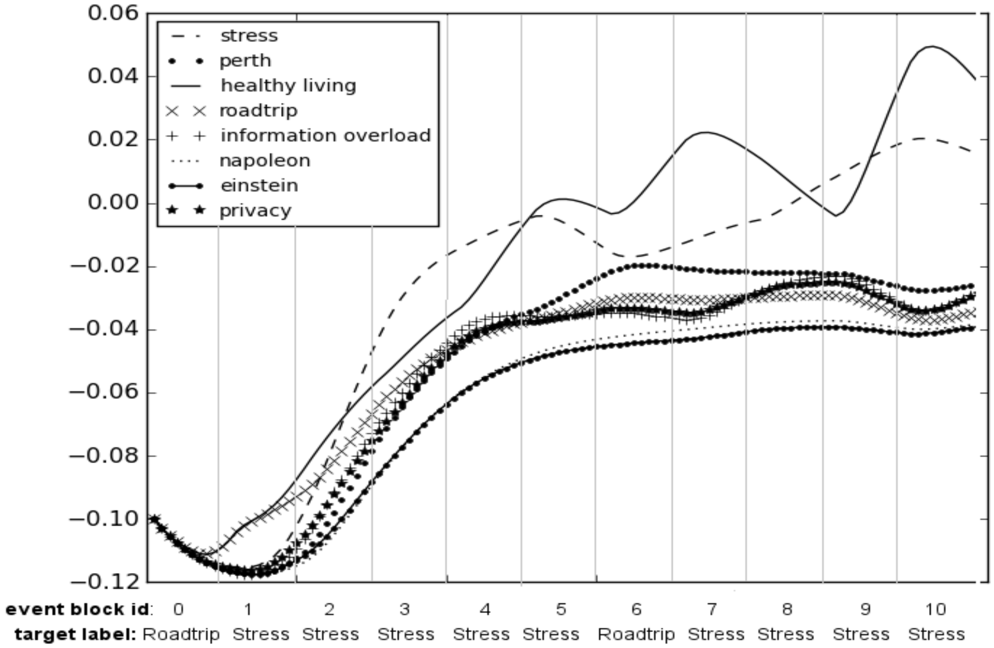

Fig. 6. Activation on the identification level with 10 iterations per event-block. The x-axis shows the id-number and the target label of the event block.

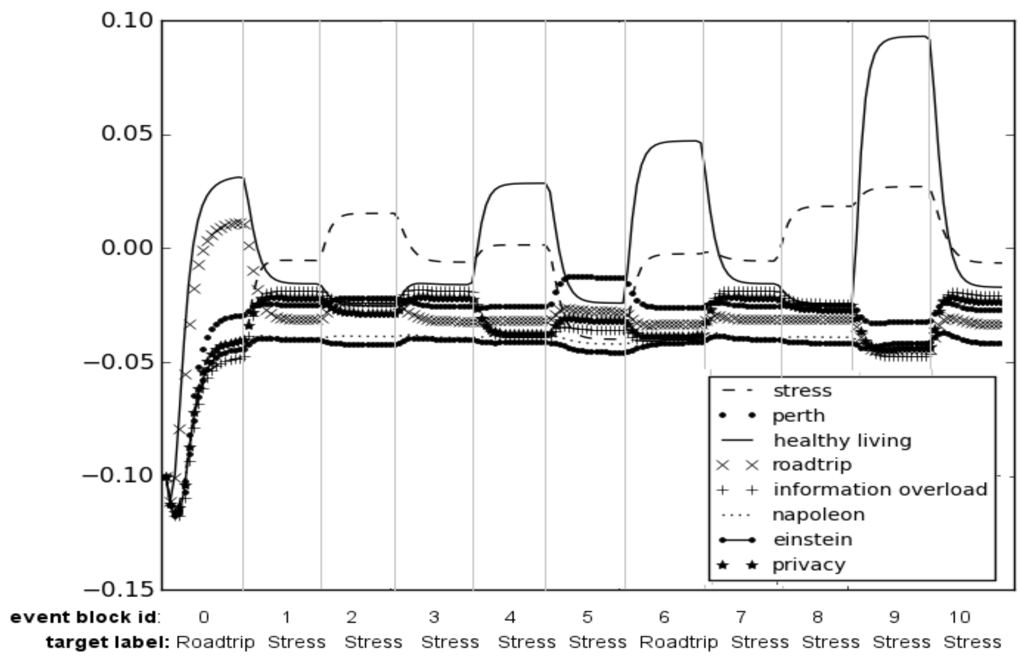

Fig. 7. Activation on the identification level with 100 iterations per event-block. The x-axis shows the id-number and the target label of the event block.

cannot conclude from one run that this is a representative outcome of the model, so we average over multiple runs.

In addition we provide results for a CIA model where the topics are based on term extraction rather than the LDA topic model (b). In this setting the topic model consists of unigrams, bigrams and trigrams that are extracted using the method described in Verberne et al. [2013]. For each of the document folders (the projects of interest) that consist of 1 Wikipedia document each, we extract the top 1000 terms after which stop-word unigrams are removed. The importance of a term is determined by comparing its frequency against its frequency in a corpus of general English (Corpus of Contemporary American English), after which hapaxes and partial terms (i.e. unigrams or bigrams that are part of another bigram or trigram in the list) are removed. In effect, each project category contains 548 salient terms on average . Examples of extracted terms are “world health organization”, “posttraumatic stress”, “american citizen”, and “dementia”. The 8 lists of salient terms are pooled together, resulting in a list of 4206 terms. These are used as nodes in the network, with the connection strengths as motivated in Table I.

The remaining results in Table IV are obtained using Weka and are used as baselines to compare CIA to. Results c. and d. originate from a k-NN classifier, with $k = 1$ (optimal k) and e. and f. originate from a Naive Bayes classifier. The results for these methods have been obtained using 10 fold cross-validation on the event-block data. This means that 90% of the event-block data was used for training and 10% for evaluations, so the type of training data is different from the training data of the CIA model. The CIA model does not need examples of labelled event block data. Baselines c. and e. receive the same raw input as the CIA model receives during run-time (e.g. window title, typed keys, caption, url), but without the additional pre-processing that CIA uses for topic and entity determination. The k-NN and Naive Bayes classifiers are, however, provided with vector representations of the full content of the document if there was a reference to a document in the event block as additional features. This is common for current approaches to context identification [Cheyer et al. 2005; Stumpf et al. 2005; Granitzer et al. 2009; Devaurs et al. 2012]. Baselines d. and f. were obtained by additional feature extraction using LDA topic recognition and entity extraction using the Stanford entity recognizer. This results in the same feature set that is used in the CIA model on the context level.

Table IV shows that the CIA network with LDA (a) has an increased performance over both k-NN (c,d) and Naive Bayes (e,f). The difference is significant in a 2-tailed t-test with $P < 0.001$ regardless of whether the improved feature extraction was used or not. The feature extraction using LDA and entity extraction does improve the quality of k-NN and Naive Bayes classifiers. The contextual IA model with term extraction (b) is significantly better than Naive Bayes (e), $P < 0.001$, but not better than k-NN (c,d) or Naive Bayes with LDA modelling (f), $P >= 0.381$. There is no significant difference between the CIA model with LDA and with term extraction.

Between the runs with the LDA model (a) there was an average spread of 10.46 percent point in accuracy over participants (average minimum 59.37%, average maximum 69.84%). This means that depending on the specific LDA model that is used, there can be a large difference in performance of the model, even when it is trained on the same data each initialization. This variation also occurs in the k-NN and Naive Bayes runs with LDA as feature selector (d and f). A possible explanation could be that this variation in performance is a result of the size of our corpus (only 8 documents), however results with larger corpora (either 4582 documents – all websites that were observed during the entire experiment – or 6561 documents – the 8 Wikipedia including their outlinks respectively) showed just as much performance variation. A repeated measures analysis where each run is seen as a measurement for a participant showed that the variation between runs was not significant ($P = 0.344$ in CIA).

Table V. Precision, Recall and F1-measure for CIA with LDA and the best k-NN and Naive Bayes baseline runs.

Run	Precision			Recall			F1-score		
	CIA	k-NN	NB	CIA	k-NN	NB	CIA	k-NN	NB
Einstein	0.32	**0.36**	0.12	**0.44**	0.27	0.32	**0.37**	0.30	0.17
Privacy	0.57	**0.66**	0.55	**0.73**	0.58	0.62	**0.64**	0.61	0.60
Information Overload	0.07	**0.27**	0.06	**0.47**	0.28	0.34	0.12	**0.27**	0.10
Roadtrip	**0.70**	0.69	0.63	0.49	**0.60**	**0.60**	0.57	**0.64**	0.61
Healthy Living	**0.66**	0.60	0.50	**0.62**	0.54	0.57	**0.64**	0.57	0.53
Perth	**0.80**	0.73	0.72	**0.78**	0.73	0.69	**0.79**	0.74	0.71
Stress	**0.67**	0.66	0.59	**0.71**	0.63	0.61	**0.69**	0.64	0.60
Napoleon	**0.87**	0.67	0.67	**0.76**	0.59	0.61	**0.81**	0.63	0.63
Average	**0.58**	**0.58**	0.48	**0.63**	0.53	0.55	**0.58**	0.55	0.49

When we look at the precision and recall values for the various classes, it is clear that some classes are more easily recognized than others. Perth and Roadtrip have high precision regardless of the classification approach, while Einstein and Information Overload have low precision. One explanation for this finding is the type of assignment that Einstein and Information Overload originate from. Both topics were short questions asked via e-mail during the experiment to distract the user rather than the assignment to write a report or prepare a presentation. Because these assignments were smaller tasks, they occur much less often in the data. Napoleon has a remarkably high precision in CIA, compared to k-NN and Naive Bayes. Overall CIA seems to have a little higher recall compared to k-NN and Naive Bayes. k-NN tends to have a higher precision than recall.

5.4.2. Effect of personal working style. The CIA approach (using LDA or term extraction) gave the best accuracy in context identification for 84% of the participants. For the remaining participants, Naive Bayes using LDA was the best approach. This shows that CIA is a robust approach that is not influenced much by personal working style, which is in contrast to the findings by Koldijk et al. [2012]

When we analyse the results of one of the participants for which both CIA using LDA performs well (participant 8, average accuracy 78.54%) and for which CIA performs poorly (participant 4, average accuracy 44.60%) there seem to be a few characteristics of the data that may have played a role. First of all, the participant 4 has fewer event blocks than participant 8 (123 compared to 475), meaning that the avarage duration of the event blocks was longer. However, we have found no significant Pearson correlation between the number of event blocks and the accuracy when we take all participants into account.

Second, participant 8 seems to be a copy-cat; he pasted text (copied from a web page to a document for example) in 20% of his event blocks, while participant 4 only did this for 5% of his event blocks. Since copied text is captured in the caption-data of the event-blocks this may have given the network a richer input compared to typed text. The typed text contains all keystrokes, including all typos and backspaces in case of corrections. It does not contain the resulting correct word, since it is not corrected for typing or spelling errors, which makes it a very noisy data source. Finally, the human annotators rated the confidence in their subject labels for participant 4 on average 3.97 on a 5 point scale with 5 being completely certain, while the annotators rated their confidence in labels for participant 8 on average 4.75. This suggests that the data for participant 4 might have been more ambiguous or unclear in general.

Another explanation is found in the majority class of the data for the participants. For participant 4, the majority class is Healthy Living, which comprises 38% of his data. For participant 8 the majority class is Perth which is 37% of his data. Since the precision for the class Perth is in general higher (0.8) than that of Healthy Living (0.65), it is likely that participant 8 will have more correct predictions than participant 4.

5.4.3. Influence of language. Two of the participants (participant 1 and 7) wrote their reports in Dutch, so we expected that their performance would increase when we trained the LDA model on Dutch equivalents of the English Wikipedia pages. Unfortunately we did not have a Dutch model for the named entity recognizer. However, because of the similarities between Dutch and English, some of the important entities (Perth, Napoleon) could still be found. Figure 8 shows the results.

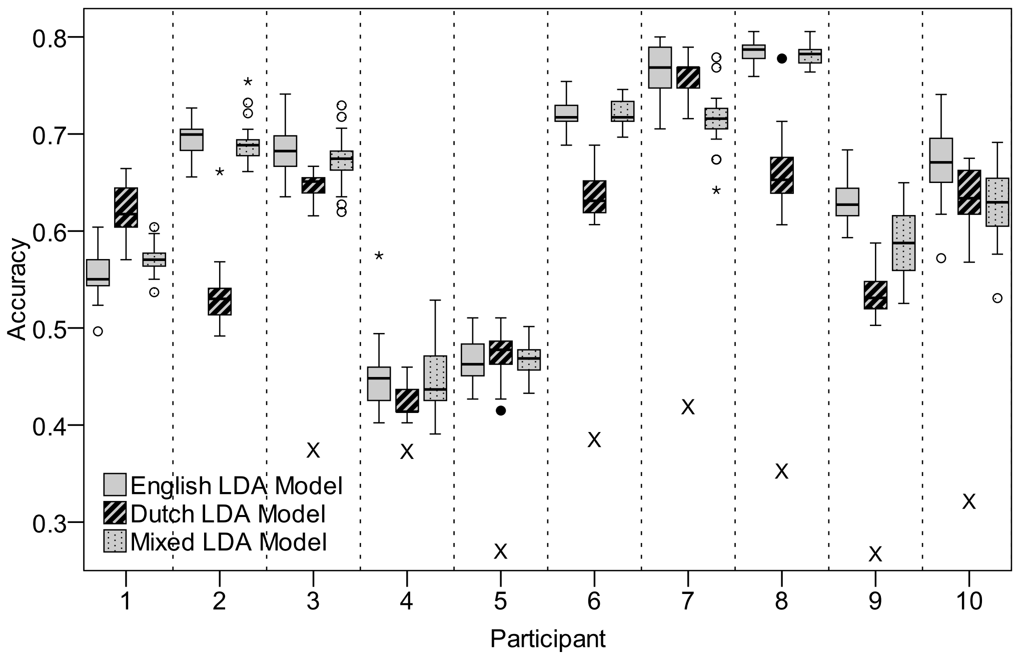

Fig. 8. Influence of language on which the LDA is trained for participants 1-10. The majority baseline for the participant is represented with X. For participants 1 and 2 the majority baseline is < 0.26 and hence not in the figure.

First of all it is interesting to see that the participants writing in Dutch do not stand out in terms of accuracy compared to the other participants. Surprisingly, the use of a Dutch LDA model lead did not have much effect on the accuracy for participant 7 (76.04% for Dutch compared to 76.59% for English). For participant 1 it lead as expected to an increase in accuracy (62.15% for Dutch compared to 55.60% for English). In total there were 4 participants that benefited from the Dutch model. There are several possible explanations why 3 of these participants benefited from the Dutch model even though they did not write their reports in English. The first explanation is that the documents in the Dutch corpus are not word by word translation of their

English equivalents, so the actual information in the Dutch corpus might be different from the information in the English corpus. In general the Dutch corpus is a bit more sparse than the English, because the documents are shorter. It may well be that the English corpus contains more irrelevant topics. Another possible explanation can be that users, even though they wrote their reports in English, visited Dutch web-pages or issued their queries in Dutch. This may have been a side-effect of the fact that the homepage of the browser during the experiment was http://www.google.nl rather than http://www.google.com

Since the participants might have used a mix of both Dutch and English (the 'Mixed LDA model' in Figure 8), we used an LDA trained on both corpora as well. In general this approach performed worse then using an English model, but slightly better than using the Dutch model. This is most likely because the model finds separate Dutch and English topics, but still has a maximum of 50 topics, so compared to the models for 1 language, this model will have less fine grained topics. The mixed approach seems to have a preference for English topics.

5.4.4. Influence of information type. One of our research questions was RQ3: 'What information is required for successful classification?'. For that reason we have run the network in several variants where we leave out either location-nodes, time-nodes, entity-nodes or topic-nodes. This provides insight in which elements are necessary for context identification and which are not so important.

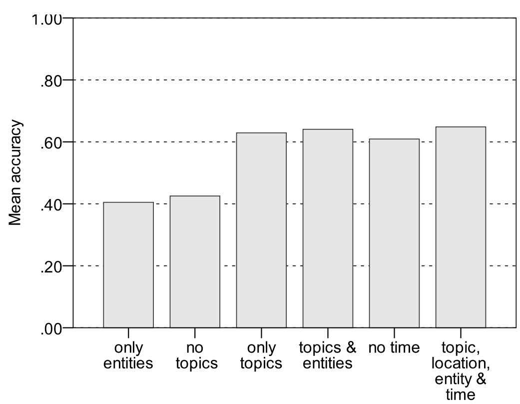

Fig. 9. Influence of information types on performance of network with LDA

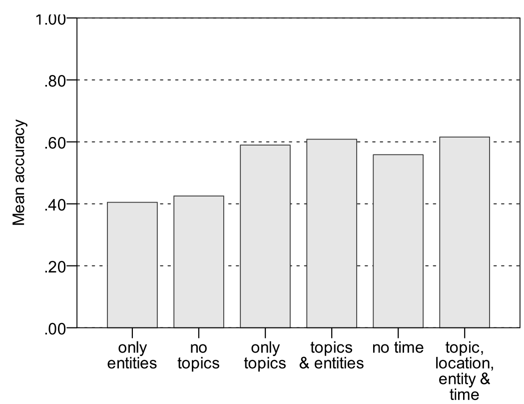

Fig. 10. Influence of information types on the network with term extraction

The average performance of the various combinations of node-types is presented in Figure 9. We see that a network with only entity nodes yields an average accuracy of 40.49%. However, only considering topic nodes, the network already has an average accuracy of 62.91% (significant improvement over just entities; $P < 0.001$). There is a slight but significant improvement ($P = 0.013$) to an average accuracy of 64.85% when adding the location, entity and time nodes compared to just topic nodes. The influence of time in the network is minimal.

Figure 10 shows the effect of using the term extraction instead of the LDA model. Again there are significant differences between the various information type combination, but now we see that a network with only term extraction gives an average accuracy of 58.96%. Again, the full network (terms, entities, locations and time) outperforms a network with just entities, or just topics ($P < 0.001$).

In both cases we see that the performance of the network is largely determined by the topic features. However, adding other information types can increase performance. Overall, time and location have little impact on the results. This is caused by the limitations of the data. We cannot estimate the influence of the time nodes realistically since each participant executed all their tasks on one day, so it will have little or no impact. Furthermore, our project-directories for training are only mock-ups and do not occur in the event-stream data, so no identification nodes can be activated based on location only, even though location is most likely strong indicator of a certain context in a realistic scenario.

5.4.5. Contextual IA model on the fly. In the previous sections we initialized the model with all the event data after which we have run the network to determine the accuracy in context detection. This is consistent with our application idea where we present the user with an overview at the end of his day of how he spent his time that day.

We can, however, imagine a scenario in which we want to inform the user about his detected context immediately (for example for recommending relevant documents) as would be the case in our 'working in context' use case. Real-time context detection could be used to categorize information objects at creation time, or to get feedback from the user about detection accuracy which can be used to improve the network, or to filter out incoming information that is irrelevant for the current context and therefore might distract the user.

In this section we show the performance of the model when run on the fly. The initial training phase then only consists of training the LDA model and making connections between identification nodes and the entities and topics that are relevant for the identification nodes. All other nodes and connections such as visited websites will be created on the fly (i.e. real-time). Note, however, that no additional connections to the context identification nodes are made, so even though the events are labelled, we do not use this information. The on-the-fly addition of the events to the network, only increases the number of associations that are made between context nodes and document nodes, and document to document connections. The real-time accuracy (presented stream-data of 0%) is presented in Figure 11. The figure also shows the delayed real-time performance (presented stream-data bigger than 0%). Thus, the accuracy of context prediction when the network has seen part of the event data already. For example, the system would start real-time prediction only after it has seen 10% of the data (about 18 minutes of data) compared to starting real-time prediction immediately (i.e. cold start). When the knowledge worker has been using CIA for a longer period of time, this can be seen as a delayed start of real-time prediction

The results show that as expected the real-time performance is significantly lower than when the model is built based on all event data ($P < 0.001$). This is because the model starts with hardly any associations in its network. With few associations, the expansion behaviour of the network using the activation function is ineffective. Real-time performance (47.09% averaged accuracy) is equal to the non real-time Naive Bayes baseline ($P = 1.000$) but worse than non real-time k-NN ($P = 0.001$). These baselines are an unfair comparison, however, as they are *not* real-time and moreover trained on 90% of the event-blocks data, whereas CIA uses no labelled event-blocks at all. k-NN and Naive Bayes would have 0% accuracy in a real-time setting as they require labelled event-block examples beforehand. Reducing the number of training examples for k-NN and Naive Bayes reduces their performance, while running the network with a delay (i.e. presenting some data to it before running) increases the network's performance. Thus overall, the network has a clear advantage in a real-time setting or when little data is available beforehand.

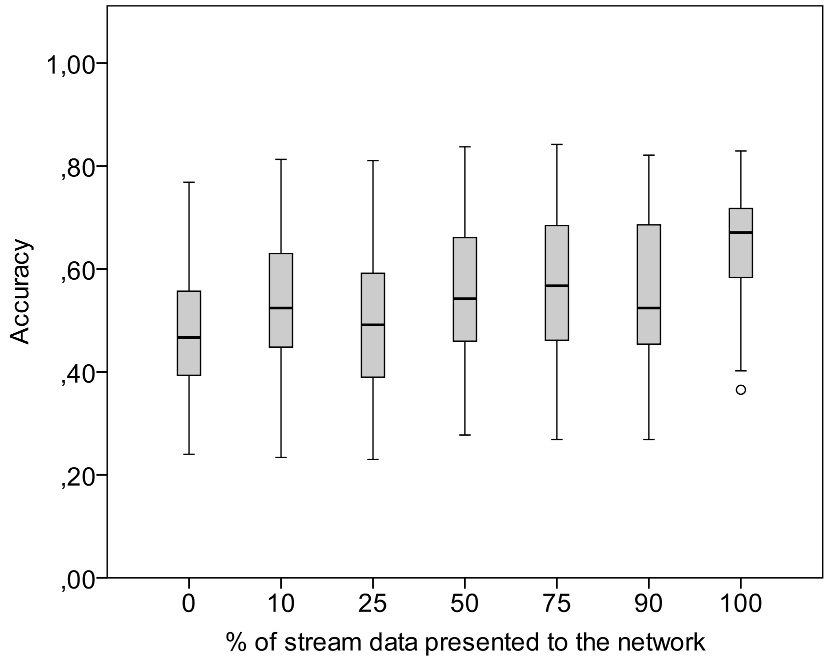

Fig. 11. Accuracy of the network with partial information presented to the model (0% represents real-time performance).

6. DISCUSSION

In this paper we presented a descriptive model for the knowledge worker's context (RQ1). The model requires the notion of a knowledge worker, resources and a possible interaction between them. Primarily the knowledge worker is engaged with a task (the knowledge worker context) and the user can be distracted or helped by interaction with resources.

The implementation of a partial version of this descriptive model, CIA, has two main advantages. The first is that it needs very little labelled data to be used for context recognition and identification (RQ2). Instead of labelled event blocks, it uses one representative document for each context that needs to be identified. In the experiment we used one document per context identification tag, which was already sufficient for the task of context identification. This equals 8 labelled documents in total for the 8 context, compared to the 8474 event blocks (90% of all available event blocks) that need to be labelled for the k-NN and Naive Bayes baselines, which is a reduction of 99% in labelling effort.

The addition of more relevant documents might improve the confidence of the connections that are being made, although this is likely to be dependent on the quality and topics in the documents. Furthermore it could be beneficial to look into the usage of documents written or saved by the user as training examples. These provide insight in the focus of the user, which may help personalize the context identification further.

A second advantage of the model is that it can expand sparse input data coming from a key logger to something more meaningful in terms of 4 types of information; locations, topics, entities and time. These are the types of information required for successful classification (RQ3). This form of expansion can also be meaningful when the input is not sparse, for example when there is direct access to the text the user is viewing, especially since it incorporates metadata such as location and time. The model can make relations between the information types using the document layer. Because the model does not map input to identification directly, it can use the information on the intermediate levels as well. This makes it possible to support the knowledge worker in several ways using a single model for his context. An example is by context-aware document recommendation which we will validate in future work.

CIA is effective in classifying the user's context and has an average accuracy of 64.85% (RQ4).The main disadvantage of the model is that, when used with an LDA model for topic extraction, the accuracy shows a lot of variation due to the non-deterministic nature of the LDA output. The improved k-NN and Naive Bayes baselines with LDA extraction suffered from the same disadvantage.

6.1. Limitations

There are some limitations in our method of evaluation. First of all, even though time and location did not contribute to the performance of the model, we can not conclude that time and location are not important for context. This is because we expect that time and location become important when data includes multiple days and repetition of activities. In our data, which is collected during three hours on a single day, location and time will have little to no impact. The results of the real-time and delayed real-time performance also shows that more data in the network is better. This may indicate that when continuing the network the next day, the performance could increase even more. Especially when you consider that the presented results are only of 3 hours of data. In future work we would like to investigate this by collecting data for multiple days of work.

Secondly, we evaluate the model on context identification, since this allows us to compare it to existing literature. However, this does not show the true purpose of the model, which lies in the possibility to use the context information layer itself. For example by using the activated context nodes as method to search or filter information. In future work we plan to address this, as soon as we can collect a suitable dataset.

6.2. Future Work

For future work, there are still many characteristics of the model to explore. First, we could use graph clustering techniques to see whether we can make the connections to identification nodes without user input.

Second, a disadvantage of the model is that it can become very complex when it has monitored a user for a while, because the number of nodes and connections in the network increases. Scalabiliy becomes an issue with more data, especially when inhibitory connections are considered. Therefore we need to investigate possibilities to optimize and clean the model regularly to make sure that it runs efficiently. One possibility is to clean up obsolete connections and nodes; elements that have little added value in the network.

Furthermore, when labelled event-block data is available, we could optimize the weights in the network in order to improve the classification accuracy.

Additionally we would like to get a more complete overview of the knowledge worker's day. This means that we would like to incorporate diary information, or GPS-sensor information. This would help recognize that the user is in a meeting or that he is travelling. One idea is to create sub-networks for different types of situations (physical activity, computer interactions, emotions, planned activities) and combine them in a larger network.

Finally, although CIA is effective, it relies on a keylogger and other content data for input. This has consequences in terms of the employability of the method in a industry setting. In essence the event data is the most privacy sensitive, as this shows how often and how long certain pages are visited. This information, however, does not need to be saved in the CIA network for it to run effectively. It is only used as activator. Without the event data the information stored in the network is not more sensitive than the saved documents on a user's computer. We envision that CIA runs as a local application on the user's working device and data structures are kept local and password protected. The output of the identification layer (after activation) can be considered sensitive, but as it is aggregated to a project level (which the user can control) the sensitivity is limited. If this data is accessible to the employer, it is important to aggregate it over multiple users to further limit the sensitivity. Other privacy preserving data mining techniques exist [Aggarwal and Philip 2008]. Future work should address the effectiveness of CIA when using such privacy preserving data mining methods.

7. CONCLUSION

In this paper we presented a new descriptive model for the context of a knowledge worker. The descriptive model was operationalized by an applied, cognitively plausible approach to context recognition and identification, which is the main contribution of this paper. This applied approach was presented as the contextual IA model (CIA-model), which is adapted from the Interactive Activation model by McClelland and Rumelhart [1981]. The model was evaluated on human computer interaction data that is representative of knowledge worker's activities. In the task of context identification the model performs at least as well as, but in general better than, k-NN and Naive Bayes baselines. Since the evaluation dataset is publicly available [Sappelli et al. 2014b], the current work can be used as a new baseline for context identification.

We summarize the advantages of CIA as follows:

(1) CIA is at least as good, but in general more successful in identifying the active context than k-NN and Naive Bayes baselines
(2) CIA tremendously reduces the required labelling effort in comparison with k-NN and Naive Bayes by using a form of transfer learning.
(3) CIA can deal with sparse and noisy inputs by making use of associations in the network to expand the input to more meaningful elements.
(4) CIA is flexible in the type of information that is represented in the context information layer, creating opportunities for many application areas.
(5) CIA is robust against differences in personal working style

The main disadvantage of CIA, is that the method for topic extraction used has a large influence on the overall performance of the model.

In the introduction we identified two use cases for which we can use the CIA model. For the proposed use case 'user-context awareness', where we provide the user with an overview of how he spent his day, or how long he spent on each activity, an average of 64.85% accuracy may not be sufficiently accurate, even though it's better than the alternative method. However, especially in the case of hour tracking in a company setting it would suffice to find estimates of longer time periods (e.g. hours). This would be an easier task than to find the correct label for each sparse event block of a few seconds separately.

For the 'working in context' use case, the identification needs to be more precise. This means that the accuracy needs to be improved. For this use case though, the context information layer and document layer could be used directly to filter information without the need of *identifying* the context first. We will explore this in ongoing research.

In future work we will evaluate the CIA-model on a real-time context-aware information delivery task (contextual support). In addition we will work on methods for automated context discovery. This would make the model more flexible, as it would create the possibility to remove and add new contexts to be identified on the fly.

ACKNOWLEDGMENTS

This publication was supported by the Dutch national program COMMIT (project P7 SWELL). Additionally we would like to thank prof. dr. ir. Th.P. van der Weide for his contributions to the descriptive model of the knowledge worker and his context.

REFERENCES

Charlie Abela, Chris Staff, and Siegfried Handschuh. 2010. Task-based user modelling for knowledge work support. In *User Modeling, Adaptation, and Personalization*. Springer, 419–422.

Charu C Aggarwal and S Yu Philip. 2008. *A general survey of privacy-preserving data mining models and algorithms*. Springer.

Varol Akman and Mehmet Surav. 1996. Steps toward formalizing context. *AI magazine* 17, 3 (1996), 55.

John R Anderson and Gordon H Bower. 1973. *Human associative memory.* VH Winston & Sons.

Liliana Ardissono and Gianni Bosio. 2012. Context-dependent awareness support in open collaboration environments. *User Modeling and User-Adapted Interaction* 22, 3 (2012), 223–254.

Marcelo G Armentano and Analía A Amandi. 2012. Modeling sequences of user actions for statistical goal recognition. *User Modeling and User-Adapted Interaction* 22, 3 (2012), 281–311.

Andrew Arnold, Ramesh Nallapati, and William W Cohen. 2007. A comparative study of methods for transductive transfer learning. In *Data Mining (ICDM)*. IEEE, 77–82.

Mohammad Taha Bahadori, Yan Liu, and Dan Zhang. 2011. Learning with minimum supervision: A general framework for transductive transfer learning. In *Data Mining (ICDM)*. IEEE, 61–70.

Travis Bauer and David B Leake. 2001. Word sieve: A method for real-time context extraction. In *Modeling and Using Context: Third International and Interdisciplinary Conference, CONTEXT*. Springer, 30–44.

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

Oliver Brdiczka. 2010. From documents to tasks: deriving user tasks from document usage patterns. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 285–288.

Patrick Brézillon. 1999a. Context in Artificial Intelligence: I. A survey of the literature. *Computers and artificial intelligence* 18 (1999), 321–340.

Patrick Brézillon. 1999b. Context in Artificial Intelligence: II. Key elements of contexts. *Computers and artificial intelligence* 18 (1999), 425–446.

Patrick Brézillon and Jean-Charles Pomerol. 2001. Modeling and using context for system development: Lessons from experience. *Journal of decision systems* 10, 2 (2001), 265–288.

A. Cheyer, J. Park, and R. Giuli. 2005. Iris: Integrate. relate. infer. share. In *Semantic Desktop Workshop*. Citeseer.

Brenda Dervin. 1997. Given a context by any other name: Methodological tools for taming the unruly beast. *Information seeking in context* 13 (1997), 38.

Didier Devaurs, Andreas S Rath, and Stefanie N Lindstaedt. 2012. Exploiting the user

interaction context for automatic task detection. *Applied Artificial Intelligence* 26, 1-2 (2012), 58–80.
Anind K Dey, Gregory D Abowd, and Daniel Salber. 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction* 16, 2 (2001), 97–166.
Ton Dijkstra, Walter JB Van Heuven, and Jonathan Grainger. 1998. Simulating cross-language competition with the bilingual interactive activation model. *Psychologica Belgica* (1998).
Paul Dourish. 2004. What we talk about when we talk about context. *Personal and ubiquitous computing* 8, 1 (2004), 19–30.
Vadim Ermolayev, C Ruiz, Marcel Tilly, Eyck Jentzsch, Jose Manuel Gomez-Perez, and Wolf-Ekkehard Matzke. 2010. A context model for knowledge workers. In *Proceedings of the 2nd Workshop on Context, Information and Ontologies*, Vol. 626. Impact Information.
Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 363–370.
J.M. Gomez-Perez, M. Grobelnik, C. Ruiz, M. Tilly, and P. Warren. 2009. Using task context to achieve effective information delivery. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*. 3.
M. Granitzer, A. S Rath, M. Kröll, C. Seifert, D. Ipsmiller, D. Devaurs, N. Weber, and S. Lindstaedt. 2009. Machine learning based work task classification. *Journal of Digital Information Management* 7, 5 (2009), 306–314.
Tudor Groza, Siegfried Handschuh, Knud Moeller, Gunnar Grimnes, Leo Sauermann, Enrico Minack, Mehdi Jazayeri, Cedric Mesnage, Gerald Reif, and Rosa Gudjonsdottir. 2007. The NEPOMUK Project-On the way to the Social Semantic Desktop. In *Proceedings of the 3rd International Conference on Semantic Technologies*. Graz University of Technology, 201–210.
Cathal Gurrin, Alan F Smeaton, and Aiden R Doherty. 2014. LifeLogging: personal big data. *Foundations and Trends in Information Retrieval* 8, 1 (2014), 1–125.
Karl Gyllstrom and Craig Soules. 2008. Seeing is retrieving: building information context from what the user sees. In *Proceedings of the 13th international conference on Intelligent user interfaces*. ACM, 189–198.
Peter Ingwersen and Kalervo Järvelin. 2005. *The turn: Integration of information seeking and retrieval in context*. Springer Science & Business Media.
Mik Kersten and Gail C Murphy. 2012. Task Context for Knowledge Workers. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.
Saskia Koldijk, Maya Sappelli, Suzan Verberne, Mark A Neerincx, and Wessel Kraaij. 2014. The SWELL Knowledge Work Dataset for Stress and User Modeling Research. In *Proceedings of the 16th International Conference on Multimodal Interaction*. ACM, 291–298.
S. Koldijk, M. van Staalduinen, M. Neerincx, and W. Kraaij. 2012. Real-time task recognition based on knowledge workers' computer activities. In *Proceedings of ECCE 2012*.
Heiko Maus, Harald Holz, Ansgar Bernardi, and Oleg Rostanin. 2005. Leveraging passive paper piles to active objects in personal knowledge spaces. In *Professional Knowledge Management*. Springer, 50–59.
Heiko Maus, Sven Schwarz, Jan Haas, and Andreas Dengel. 2011. CONTASK: Context-Sensitive Task Assistance in the Semantic Desktop. In *Enterprise Information Systems*. Springer, 177–192.
Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language

Toolkit. (2002). http://mallet.cs.umass.edu.
John McCarthy. 1993. Notes on formalizing context. (1993).
James L McClelland and David E Rumelhart. 1981. An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological review* 88, 5 (1981), 375.
N. Oliver, G. Smith, C. Thakkar, and A. C Surendran. 2006. SWISH: semantic analysis of window titles and switching history. In *Proceedings of the 11th international conference on Intelligent user interfaces*. ACM, 194–201.
Masaki Omata, Kei Ogasawara, and Atsumi Imamiya. 2010. A project restarting support system using the historical log of a userś window usage. In *Proceedings of the 22nd SIGCHI conference*. ACM, 25–32.
Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.
Carlo Penco. 1999. Objective and cognitive context. In *Modeling and Using Context*. Springer, 270–283.
Andreas S Rath, Didier Devaurs, and Stefanie N Lindstaedt. 2010. Studying the factors influencing automatic user task detection on the computer desktop. In *Sustaining TEL: From Innovation to Learning and Practice*. Springer, 292–307.
M. Sappelli, S. Verberne, and W. Kraaij. 2012. Using file system content to organize e-mail. In *Proceedings of the fourth symposium on Information interaction in context*. ACM.
M. Sappelli, S. Verberne, and W. Kraaij. 2014a. E-mail categorization using partially related training examples. In *Proceedings of the fifth symposium on Information interaction in context*. ACM.
M. Sappelli, S. Verberne, Koldijk S.J., and W. Kraaij. 2014b. Collecting a dataset of information behaviour in context. In *Proceedings of the 4th Workshop on Context-awareness in Retrieval and Recommendation*.
Leo Sauermann. 2009. *The gnowsis semantic desktop approach to personal information management*. Ph.D. Dissertation. Ph. D. thesis, Fachbereich Informatik der Universität Kaiserslautern.
Benedikt Schmidt, Todor Stoitsev, and Max Mühlhäuser. 2011. Task Models for Intention-Aware Systems. *J. UCS* 17, 10 (2011), 1511–1526.
J. Shen, L. Li, T. G Dietterich, and J. L Herlocker. 2006. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th international conference on Intelligent user interfaces*. ACM, 86–92.
Tadej Štajner, Dunja Mladenić, and Marko Grobelnik. 2010. Exploring contexts and actions in knowledge processes. In *Proceedings of the 2nd International Workshop on Context, Information and Ontologies*.
Simone Stumpf, Xinlong Bao, Anton Dragunov, Thomas G Dietterich, Jon Herlocker, Kevin Johnsrude, Lida Li, and J Shen. 2005. Predicting user tasks: I know what youŕe doing. In *20th National Conference on Artificial Intelligence, Workshop on Human Comprehensible Machine Learning*.
S. Verberne, M. Sappelli, and W. Kraaij. 2013. Term extraction for user profiling: evaluation by the user. In *Proceedings of the 21th International Conference on User Modeling, Adaptation and Personalization*.
Paul Warren. 2013. Personal Information Management: The Case for an Evolutionary Approach. *Interacting with Computers* (2013).
P. Warren, J. M Gomez-Perez, C. Ruiz, J. Davies, I. Thurlow, and I. Dolinsek. 2010. Context as a tool for organizing and sharing knowledge. In *Proceedings of the 2nd International Workshop on Context, Information and Ontologies*.
Stewart Whiting and Joemon Jose. 2011. Context modelling for situation-sensitive

recommendations. In *Flexible Query Answering Systems*. Springer, 376–387.