A Calendar and To-Do List with Common Sense

Henry Lieberman, Alexander Faaborg, Jose Espinosa, and Chris Tsai

MIT Media Laboratory, 20 Ames St., Building E15, Cambridge, MA 02139 USA
lieber@media.mit.edu, faaborg@media.mit.edu, jhe@media.mit.edu,
cktsai@mit.edu

Abstract. People often rely on the to-do lists integrated into their calendar applications to remember basic tasks they must complete during a hectic day. However, current calendar applications require users to enter every task into their to-do list manually. But each event in life is often connected to others; some events require prerequisites, others imply subsequent events. Entering tasks manually is unreliable, because busy users often forget to enter all events related to a single goal or purpose. We propose an approach for automatically suggesting related events to an event entered by the user, based on Commonsense knowledge. It is based on LifeNet, a semantic network of partially ordered events in everyday life, generated from Open Mind Common Sense, a collection of 700,000 English statements contributed by Web volunteers. This approach results in reduced cognitive load on users, and a streamlined user interface.

1 Introduction

The ideal calendar interface would be as intelligent as a human assistant. It would politely remind you to buy orange juice if you were sick, and to remember to pack sunscreen before going to the beach. It would remind you to make a dinner reservation before your anniversary, and to send a thank you note after a job interview. All of these activities are common sense, but they are also things people are prone to regularly forget. Because of this, users often rely on Personal Information Management applications like Microsoft Outlook and Lotus Organizer. These applications allow users to delegate remembering a task to a computer's memory, instead of their own memory. However, this delegation only solves part of the problem, as users must remember to input the task in the first place. The reason current generation Personal Information Management applications cannot proactively suggest tasks is because they know very little about their users, and the world they inhabit.

The field of human-computer interaction has predominantly focused on improving usability by simplifying user interfaces, making it easier for humans to understand computers. However, progress can also be made by taking the opposite approach: improving usability by making it easier for computers to understand humans. Both of these approaches have been applied to improving calendar applications. Previous research in calendar applications has explored new types of user interfaces and new approaches information visualization, but it has also focused on new techniques for modeling user's behavior. The present work fall squarely in this category, using models of everyday life to suggest tasks related to the user's appointments.

2 Using Commonsense Reasoning to Anticipate User Tasks

People use calendars to help them manage their lives. But life is not a series of totally disconnected events. Life events are related to each other in a variety of ways. Some events must occur before others can take place. You must by an airplane ticket before you can fly to another city. Some events require resources that must be obtained. Since buying an airplane ticket requires money, it is possible some steps may need to be taken to obtain the money necessary to pay for a ticket. Some events imply that others will take place in the future. If you take a business trip, you must later submit a request to be reimbursed. Events have purposes. You might schedule a meeting with a colleague to work on a problem, but a phone call or e-mail message may be substitutable for the meeting. Some events need to happen in a certain order, others can happen in parallel.

Today's commercial calendar programs provide few facilities for managing event sequences. Users are reduced to entering each event manually, a process prone to error. It is easy to enter an event, and forget to enter preceding events necessary to allow the main event to take place, or follow-up events subsequent to the main event. It is easy to forget what resources are needed or what the implications of an event are. We seek to implement an assistant that watches users enter events, and automatically produce suggestions of other, related events to be entered into the calendar.

Modeling calendar events, therefore, boils down to modeling life events. Previous work on intelligent calendars predefines a set of representations for common types of events such as meetings, the following example from [1]:

request-5-27-192-48:
attendees:thrun
event-type:meeting
date: (29 5 1992)
time: 1430
duration: 30
location: weh5309
confirmed?: yes
displayed-week:(25 5 1992)
action-time:2915977709
action-date:(27 5 192)
previous-request: request-527-1992-13
previous-prompt: confirmed=yes
position-attendees: project-scientist
previous-attendees-meeting: request-5-20-1992-1

next-attendees-meeting: none
lunchtime?:no
number-of-attendees: 1
cmu-attendees?: yes
attendees-in-toms-group?: yes
known-attendess?: yes
day-in-week: friday
endtime: 1500
busyness-ofattendees: 2
single-attendee?: yes

The system then provides a learning algorithm that can learn rules relating events, and can then anticipate future events. The problem is that an ontology of meetings or other kinds of events, must be developed, and each of these representations must typically be crafted by hand.

For a broadly applicable assistant, where the types and kinds of events might not be explicitly known in advance by the developer, it is necessary to have a broad-spectrum model of everyday life events. Such a general model of everyday life might seem like an impossibly difficult task, but recent work in our group and elsewhere, on the problem of Common Sense Reasoning, has resulted in some usable, if still incomplete, models of everyday life.

The next sections introduce Open Mind, our knowledge base about Common Sense, and its derivative LifeNet, which explicitly addresses the issue of life events and their interrelationships. This forms the basis of our calendar agent, which can recognize a broad spectrum of everyday life events without additional explicit predefinition of event types and properties by the developer of the agent.

2.1 Open Mind: Teaching Computers the Stuff we All Know

Since the fall of 2000 the MIT Media Lab has been collecting commonsense facts from the general public through a Web site called Open Mind [2,3]. At the time of this writing, the Open Mind Common Sense Project has collected over 738,000 facts from over 15,00 participants. These facts are submitted by users as natural language statements of the form "tennis is a sport" and "playing tennis requires a tennis racket." While Open Mind does not contain a complete set of all the common sense knowledge found in the world, its knowledge base is sufficiently large enough to be useful in real world applications.

Using natural language processing, the Open Mind knowledge base was mined to create ConceptNet [4], a large-scale semantic network currently containing over 1.6 million assertions. ConceptNet consists of machine-readable logical predicates of the form: [IsA "tennis" "sport"] and [EventForGoalEvent "play tennis" "have racket"]. ConceptNet is similar to WordNet [5] in that it is a large semantic network of concepts, however ConceptNet contains everyday knowledge about the world, while WordNet follows a more formal and taxonomic structure. For instance, WordNet would identify a dog as a type of canine, which is a type of carnivore, which is a kind of placental mammal. ConceptNet identifies a dog as a type of pet [4].

A subset of ConceptNet was transformed to create LifeNet [6,7]. LifeNet is a large-scale temporal graphical model expressed in terms of egocentric propositions of the form: [”I buy present” - ”I go to birthday party”]. Presently LifeNet consists of a total of 80,000 propositional nodes linked by 415,000 joint probability tables between pairs of nodes.

2.2 Creating LifeNet

From ConceptNet a graph of temporal relations called LifeNet [6,7] was created. To create LifeNet, first we selected the action, state and object nodes, and the temporal related links from ConceptNet. In addition, we used the Open Mind corpus to generate a correlation file of the frequency of each pair of words. ConceptNet nodes were paraphrased to be egocentric. Then, we created rules that map between the egocentric nodes, using the temporal ConceptNet relations. Finally, we used the correlation data to produce a probability represented the confidence of the relationship. More information about this process is contained in ”LifeNet: A Propositional Model of Ordinary Human Activity” [6].

We use spreading activation to perform inference over this network. The first operation is to find the temporally relevant events of the user’s actions, and the second is to find a plausible path between two events.

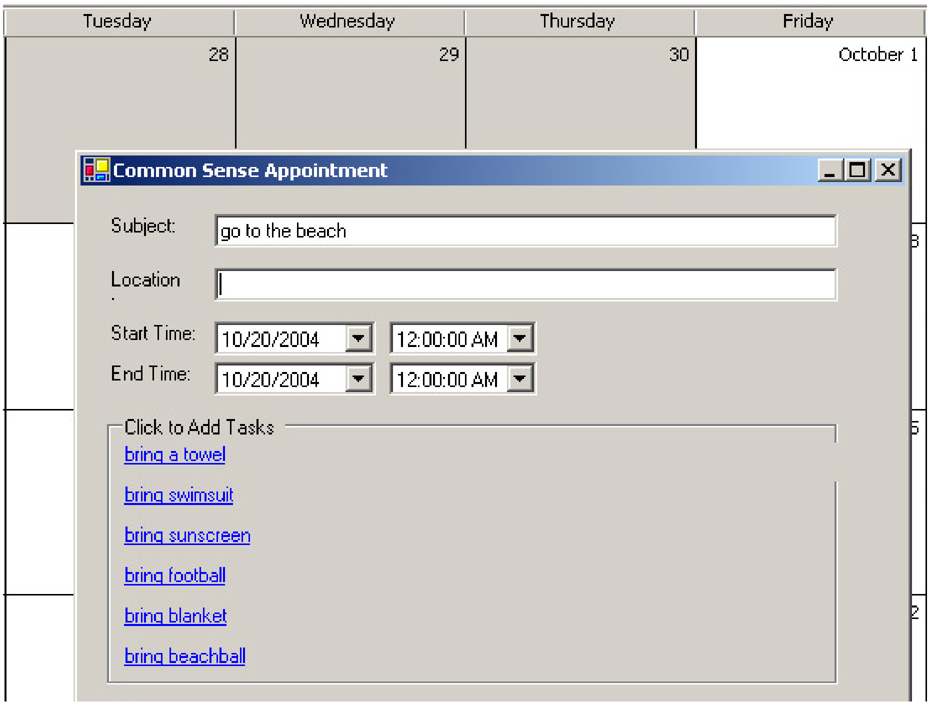
To find temporally relevant nodes, first the observed nodes are excited with a fixed amount of energy. Then, a fraction of this energy is spread to the adjacent nodes. The amount of energy transferred is proportional to the probability of the link. This step can be repeated until a certain criterion is reached. In addition, we can chose between using just spreading energy among forward links (future events), backward links (past events) or both (related events). Each step made by the algorithm can be seen as a step away in the action. The farther a node is from the origin node, the farther in time the event is. For example, the future steps of ”ride a bike” are ”I get physical activity”, ”I stay in a city, ”I work”, ”I go to a store”, ”I get in shape”, ”I exercise”, ”I get exercise”, ”I get to a other side”, ”I travel from one point to another”, ”I view scenery”, ”I keep fit”, ”I get fit”, ”I run an errand”.

Finding the plausible path between two events is inspired by Maes’s ”How to Do the Right Thing” [8]. First, the initial event is set active and exited with a positive value; the goal event with a negative value. Then, the energy is spread forward from the initial node, and backward from the goal event. A node is set to active if its energy is above a certain threshold and at least one of his preceding nodes is active and there is no node that fulfills these requirements with higher energy. The algorithm ends when the goal node is activated. Finding a path using this algorithm will promote the highly connected nodes over the less connected nodes. Since the coverage of the links is sparse, sometime there is no short link between the initial and the goal node. In this case, a new kind of link is calculated on the fly. This link uses simple keyword matching to look for nodes with similar meaning that the excited nodes. This kind of node rather meaning a next temporal step, mean different ways to express an action.

For anticipating user tasks we choose spreading activation over Bayesian algorithms since each step in the spreading activation algorithm is a temporal step and will not make much difference from Bayesian algorithms. At the time of writing this paper, we are comparing the current spreading activation versus a variety of Bayesian algorithms.

2.3 User Interface

Our user interface integrates with Microsoft Outlook, replacing the "New Appointment" dialog box. When a user creates a new appointment in their calendar, our application queries LifeNet for events that are likely to occur before and after the appointment. The application then filters the information returned from LifeNet, and automatically displays a list of semantically related tasks in real time. For instance, in Figure 1 the user has created the appointment "go to the beach", and the interface suggests the related tasks of "bring a towel, bring swimsuit, bring sunscreen" The user can then click on one of these tasks, and it will be automatically to their task list in Outlook.



2.4 Implementation

The application was writing in C# and uses the Microsoft Outlook 2003 Integration API to create new appointments and tasks. The interface shown in Figure

1 is accessible by a button that replaces the "New Appointment Button" in the Outlook toolbar, created using a COM Add-in.

The information contained in LifeNet is expressed in terms of egocentric propositions. For instance, here are the first 7 temporal connections in LifeNet to the event "go to the beach."

(0.767 "I fly my kite" "I run at a beach" 11711 28498)
(0.669 "I walk on sand" "I go to a beach" 36483 14375)
(0.669 "I relax on a beach" "I enjoy a day" 27506 9199)
(0.669 "I go to a beach" "I play frisbee" 14375 24382)
(0.669 "I find my way to a beach" "I surf" 11213 33211)
(0.611 "I sit under an umbrella" "I go to a beach" 30900 14375)
(0.611 "I collect sea shells" "I go to a beach" 4881 14375)

While all of these statements are true, none of them represent actions that users are likely to put on their to-do list. To generate tasks that users are likely to add to their to-do list, we filter the results from LifeNet against a list of common "task verbs." We determined the most common "task verbs" to be: buy, prepare, choose, reserve, decide, pick, purchase, take, bring, find, go to, and get. Through a series of trials, we determined that the application produces the most relevant potential tasks when events from LifeNet are ranked into tiers:

Tier 1: Buy
Tier 2: Prepare, Choose, Reserve, Decide, Purchase
Tier 3: Take, Bring, Find, Go To
Tier 4: Get

After filtering the results from LifeNet through this process, the top 12 potential tasks are displayed to the user.

3 Evaluation

To evaluate the effectiveness of our approach, we selected 7 example calendar appointments and analyzed the relevancy of the automatically generated tasks. The example calendar appointments we tested were:

(1) Play a game of tennis
(2) Go to the beach
(3) Eat dinner
(4) Go to a movie theater
(5) Go on vacation
(6) Go to a party
(7) Fly to a different city

On average, the application produced 3.7 relevant tasks per event. To compare these results to the types of tasks human subjects would produce, we asked 8 subjects to generate a list of 1-5 tasks for each of the 7 events. The 8 subjects produced an average of 3.8 tasks per event. Interestingly, only 40% of the subject's responses were identical to the automatically generated tasks. This means that on average, 60% of the relevant automatically generated tasks were tasks subjects didn't immediately remember, which are the best kind of suggestions.

For instance, only 1 of the 8 subjects produced the task "bring swimsuit" for the event "go to the beach." And no subjects produced the task "bring blanket" for the event "go to the beach."

Of the relevant automatically generated tasks, 84% were stated by at least 1 subject. The fact that only 40% of the subject's tasks were identical to the automatically generated tasks, but 84% were stated by at least 1 subject implies that for these test events, LifeNet's coverage was broad enough, but it was not deep enough. The subjects produced many similar tasks that our application did not predict. For instance, 5 of the 8 subjects produced the task "check weather" for the event "go to the beach." 3 of the 8 subjects produced the task "make sure tennis cothes are clean" for the event "play tennis." And 4 of the 8 subjects produced the task "pack" for the event "go on vacation." As the Open Mind knowledgebase grows, many of these common tasks should begin to appear. After the evaluation, all of the missing commonsense facts were added to Open Mind.

4 Related Work

Previous research in calendar applications can be categorized into (1) improving the way users understand their calendars, and (2) improving the way calendars understand their users. Our research falls in the latter category.

4.1 User Interfaces and Information Visualization

New calendar user interfaces include DataLens [9], a fisheye representation and a zoom-able interface to support complex scheduling tasks. Mackinlay [10] used 3D graphics to improve the visualization of large amounts of time-based information. Past research has also focused on studying the use of calendars in large organizations, including work by Palen [11] and Grudin [12].

4.2 User Modeling

The use of groupware calendar systems led to research in using Bayesian networks to predict user's attendance at events. Examples of this include Tullio's Augur calendar system [13], and Mynatt's Ambush calendar system [14]. These systems are similar to Horvitz's Lumiere Project [15], which used Bayesian networks to model user's goals and needs. Machine learning was also used by Kozierok [16] and Mitchell, et. al. [1] in a calendar application to build an interface agent that assisted users with scheduling meetings, and learned through observation.

The closest work to ours is Erik Mueller's [17] SensiCal, a calendar application that also uses Commonsense Knowledge. SensiCal uses Commonsense Knowledge from Mueller's own handcrafted knowledge base (about 100,000 items) to fill out missing information and provide intelligent defaults, like the event of dinner lasting 2 hours. SensiCal also uses Commonsense Knowledge to warn users of (but not prohibit) potential problems, like scheduling breakfast at 3am, scheduling an appointment when the user is out of town, or scheduling an appointment to take a vegetarian to a steakhouse. SensiCal [17] shares our goal of using Commonsense Knowledge to create a more intelligent and humane calendar interface. Like the other works cited, SensiCal is mainly concerned with single events at a time, rather than the connection between related events.

5 Discussion and Future Work

Like many of the other applications we have designed using Commonsense Reasoning [18], the design of the calendar agent keeps in mind the principal of fail-soft design. Because we know that the heuristics we are using are not completely reliable, we design the machine's predictions as suggested adjuncts to the user's normal operation of the interface rather than as a substitute for the user's judgment. The user is free to ignore suggestions and stick to manual additions to calendar entries if he or she so chooses.

Even taking into account that the machine's suggestions are optional, the agent could still be detrimental if the user felt that the machine's suggestions were distracting. Fortunately, we have not experienced those kinds of negative reactions in informal user testing. No one felt, for example, that the suggestions were distracting in the same way as the infamous Microsoft paper clip. Part of the reason for this is that the Commonsense Knowledge base usually comes up with plausible suggestions, even if it does not come up with correct suggestions. Compared to statistical prediction algorithms, which sometimes come up with what look like arbitrary suggestions based on accidental or hidden-variable correlations, Commonsense-based prediction makes better mistakes. The power of making better mistakes is not to be underestimated as a factor in user satisfaction.

We are far from fully exploiting the power of Common Sense in predictive interfaces for the calendar application. First, although the present implementation predicts additional event types, it does not attempt to project attributes of those events, such as the duration of event, participants, location, resources, etc. Some other projects cited above, such as Kozierok, Mitchell, Mueller, etc. do use machine-learning prediction as intelligent defaults for event attributes, and we could do so as well. Beyond what is done in these projects, we could also coordinate attribute values across related events, such as connecting dates of hotel stays to dates of flight arrival and departure times. We could also have a richer vocabulary of event relations, distinguishing between sets of events that can happen in parallel versus those that need to happen in serial, required, optional, and satisfying events.

We could also explore connections between the calendar and other applications, either desktop applications or Web applications. In a related project [18], we are exploring the world of Semantic Web services, and using Common Sense to compose sets of Web services to accomplish a particular user goal. When those Web procedures and services extend over time, integration with a calendar would be beneficial. For example, arranging a series of doctors' appointments to deal with a particular medical condition could be coordinated with Web services for medical referral, transportation, insurance, and other services.

We all could use a little help in managing our ever-increasingly hectic, busy and complicated lives. The task of keeping a calendar, as helpful as calendars are, shouldn't be yet one more opportunity to forget to do something important. A little bit of Common Sense can go a long way in helping us remember the things we need to do.

Acknowledgments

The authors would like to thank Push Singh and Hugo Liu for their helpful feedback and for breaking new ground with Open Mind Common Sense, LifeNet and ConceptNet.

References

1. Mitchell, T.M., R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, Experience With a Learning Personal Assistant. Communications of the ACM , Vol. 37, No. 7, pp. 81-91, July 1994
2. Singh, P.: The Open Mind Common Sense Project. KurzweilAI.net: http://www.kurzweilai.net/meme/frame.html?main=/articles/art0371.html (2002)
3. Singh, P.: The Public Acquisition of Commonsense Knowledge. Proc. 2002 AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access (2002) 47-52
4. Liu, H., Singh, P.: ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal. Volume 22, Number 4. Kluwer. (2004) 211-226
5. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA, USA (1998)
6. Singh, P., Williams, W.: LifeNet: A Propositional Model of Ordinary Human Activity. Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP) at KCAP (2003)
7. Singh, P., Barry, B., Liu, H.: Teaching Machines about Everyday Life. BT Technology Journal. Volume 22, Number 4. Kluwer. (2004) 227-240
8. Maes, P.: How to Do the Right Thing. Connection Science Journal, Volume 1, Number. 3 (1989)
9. Bederson, B., Clamage, A., Czerwinski, M., Robertson, G.: DateLens: A Fisheye Calendar Interface for PDAs. ACM Transactions on Computer-Human Interaction (TOCHI). Volume 11, Issue 1 (2004) 90-119
10. Mackinlay, J., Robertson, G., DeLine, R.: Developing Calendar Visualizers for the Information Visualizer. Proceedings of the 7th annual ACM symposium on User interface software and technology (UIST). Marina del Rey, California, United States (1994) 109-118
11. Palen, L.: Social, Individual and Technological Issues for Groupware Calendar Systems. Conference on Human Factors in Computing Systems (CHI). Pittsburgh, Pennsylvania, United States (1999) 17-24
12. Grudin, J.: A Case Study of Calendar Use in an Organization. ACM SIGOIS Bulletin. Volume 17, Issue 3 (1996) 49-51

13. Tullio, J., Goecks, J., Mynatt, E., Nguyen, D.: Augmenting Shared Personal Calendars. Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST). Paris, France (2002) 11-20
14. Mynatt, E., Tullio, J.: Inferring Calendar Event Attendance. Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI). Santa Fe, New Mexico, United States (2001) 121-128
15. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, K.: The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Proceedings of UAI (1998)
16. Kozierok, R., Maes, P.: A Learning Interface Agent for Scheduling Meetings. Proceedings of the 1st International Conference on Intelligent User Interfaces (IUI). Orlando, Florida, United States (1993) 81-88
17. Mueller, E.: A Calendar with Common Sense. Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI), New Orleans, Louisiana, United States (2000) 198-201
18. Lieberman, H., H. Liu, P. Singh, B. Barry.: Beating Common Sense into Interactive Applications, AI Magazine, Winter 2005.