Technical Disclosure Commons

Defensive Publications Series

January 08, 2015

Smart Date and Time Suggestions for Calendaring

Ibrahim Badr

Piotr Malinowski

Nicholas Chornay

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Badr, Ibrahim; Malinowski, Piotr; and Chornay, Nicholas, "Smart Date and Time Suggestions for Calendaring", Technical Disclosure Commons, (January 08, 2015)
http://www.tdcommons.org/dpubs_series/1

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Smart Date and Time Suggestions for Calendaring

ABSTRACT

A date and time suggestion system is described for autocompleting and suggesting dates and times in an application, for example, a calendar application. The system receives a text input for creating a calendar event from a user of the system. The text input can be an alphabet, a number, combination of letters etc, such as “t”, “to” or “9”, inputted by the user while setting up the calendar event. After receiving the text input, the system resolves the input to generate one or more candidate dates and time for the calendar event. Subsequently, the system scores the candidate dates and times to generate a ranked list of suggested dates and times for the calendar event. The scoring of the suggested dates and times help to provide more relevant results to the user. The system then provides the suggested dates and times to the user based on the ranking made by the system.

PROBLEM STATEMENT

Calendar application are routinely used to set up meeting requests, reminders, allocate time for events, etc. Present calendar applications require the user has to enter all information necessary to manually create a calendar event. This could include inputting the title, time of the event, day of the event, etc. Some calendar applications include a relatively simple “find a time” feature that identifies free time slots in the event creator’s and the invitees’ schedules. However, having to manually fill out fields and fields of information when creating calendar events can be cumbersome and repetitive. A method and system that assists the user in creating calendar events is described.

DATE AND TIME SUGGESTION SYSTEM

The systems and techniques described in this disclosure relate to a date and time suggestion system that suggests dates and times in an application, for example, a calendar application. The date and time suggestion system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented locally on a client device or implemented across a client device and server environment.

FIG. 1 illustrates an example method 100 for suggesting dates and times in a calendar application. As described below, method 100 can be performed by the date and time suggestion system.

The system receives a text input for scheduling an event (block 110) in a calendar application. The text input can include a number, e.g., '12,' or a letter, e.g., 't,' inputted by the user to create the calendar event. The calendar event being created by the user may be for a particular event, such as, a meeting, dinner, party, at a specified date and time. The text input by the user reflects the date and time at which the user wants to schedule the event. The user can use input devices, e.g., touch screen, mouse, keyboard etc., at a client device to input the text into a user interface provided by the system to set up the calendar event.

The system resolves the text input to generate one or more suggested dates and times for the event (block 120). As the user starts typing the date and/or time for the event to be scheduled, the system resolves the input to generate date and time candidates. As an example, if the user inputs 't,' the system can generate a list of possible candidates based on input 't,' such as "Thursday," "Tuesday," "tomorrow," "today," etc. Similarly, if the user inputs '12', the system can generate a list of possible candidates such as "12pm," "12am," "12.30pm," "12 April,"

"December," etc. The system can utilize a text parser that is capable of recognizing and understanding day/date and/or time mentioned in a sentence. The system uses the parser to resolve the text input inputted by the user to a list of candidate dates and times. The parser can also work in combination with the server to generate certain words and phrases that specify a time. These words are then resolved into the correct time. As an example, the system can autocomplete an input 't' into possible words such as "tomorrow", "today", "thursday." These words can then be resolved into time such as "July 4, 2014" for "tomorrow" or "July 3, 2014" into "Today" and so on. Additionally, the parser can also recognize dates of holidays. For example, a text input of 'Mother's day' by the user while setting up the calendar event can generate a possible candidate as "May 10."

The system then scores the candidate dates and times to generate a ranked list of candidate dates and times for the event to be scheduled (block 130). This scoring helps the system to understand whether "to" refers to "tomorrow" or "today" or whether "12" refers to "12pm, 12am, 12 april, or December." The system can score the resolved list by using an algorithm which considers various factors such as words in the title, how far in the future the suggestion is, date/time the event was created, free/busy times on the user's schedule, what part of the timeline the user was looking at when he tapped to create the event, information from other sources for location and so on. Additionally, or alternatively, the scoring algorithm can also utilize rules to score the candidate dates and times. One example rule can be to assume it's an all day event if only a date has been specified. Another example rule can be to assume that "this Monday" usually means the closest future date with that name and thus can imply the closest Monday in the future or that "next Monday" usually means the following Monday after this

Monday. Another example can be, when the user inputs “dinner at 6pm” and it is currently 9pm, the dinner is set for 6pm tomorrow. Another rule for the scoring algorithm can be that it should know where in the timeline the user was when he created an event and when it makes sense using the date that gets pre populated when the user creates the event. The same rule can be applicable for the date and time pre-loaded when the user creates the event from a time grid. Additionally, the scoring algorithm can also understand whether “2/3/2014” is in February or March by using different parameters such as, user time setting and location, how far the event is in the future, etc.

Consequently, the system provides the candidate dates and times based on the ranking to the user (block 140). After the system has scored the resolved list of candidate dates and times, the system can display the candidate dates and times to the user as autocompletions to the user’s text input. The autocompletions can be presented in a list order based on their respective rankings. For example, the highest ranked date and time candidate is presented as the first autocompletion of the input followed by lower ranked date and time candidates. Additionally, the system can provide a predetermined number of candidate dates and times as autocompletions to the user’s text input. For example, the system only provides the top five ranked candidate dates and terms as autocompletions to the user’s text input. The user can select one of the provided autocompletions to complete his text input. Additionally, the system can also suggest the duration and end date of an event. This duration would be appropriate to the type of the event scheduled. For example, dinner would suggest 1 or 2 or 3 hour durations while meetings would suggest 30mins or 60mins and so on.

In order to highlight the functionality of the date and the time suggestion system, consider an example where the system receives an input “dinner at yyy hotel on s...” to create an event on the calendar application utilizing the date and time suggestion system (see block 210). Now, the system will use the built in parser and/or server to generate a list of possible candidate dates and times for the user (see block 220). The system would be aware about the current day of the week when the user is creating the event. With the assumption that the current day is Thursday, the system would also be aware of other related scheduling information, such as most dinners scheduled on a thursday occur on the weekend, or that the user is busy on friday night as indicated by the user’s calendar, and/or that yyy hotel closes at 10 pm and that the user usually has his dinner at 7 pm. The system then scores the candidate dates and times to generate a ranked list of candidate dates and times for the event (see block 230). After the system has scored the resolved list of candidate dates and times, the system can display the candidate dates and times to the user as autocompletions to the user’s text input. The autocompletions can be presented in a list order based on their respective rankings. Thus, the first suggestion in the list provided by the system can be “saturday night at 7 pm” (see block 240). Additionally or alternatively, the user can select the different suggestions provided based on the ranking to correct it to a different time or day as suggested in the list by the system.

The subject matter described in this disclosure can be implemented in software and/or hardware (for example, computers, circuits, or processors). The subject matter can be implemented on a single device or across multiple devices (for example, a client device and a server device). Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user (for example, from a

mouse, keyboard, or touchscreen) and produce an output to a user (for example, through a display). Specific examples disclosed are provided for illustrative purposes and do not limit the scope of the disclosure.

Drawings

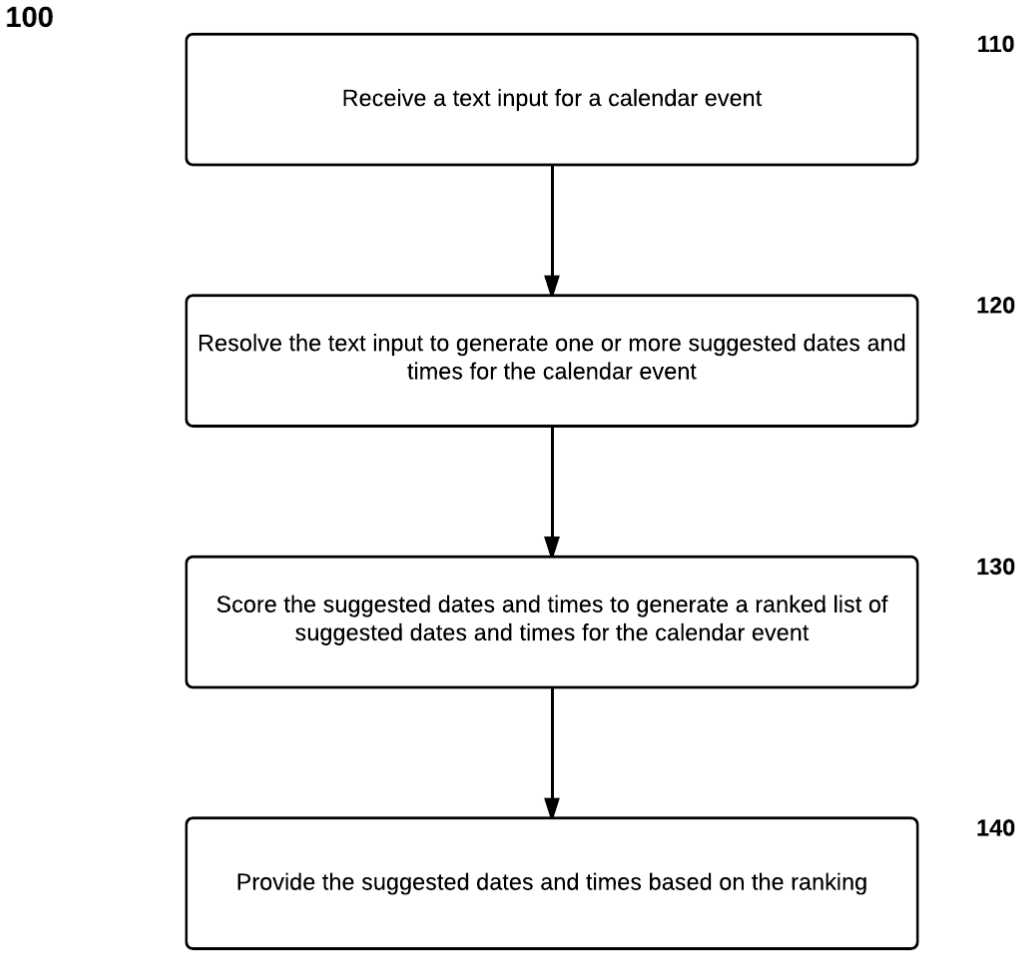
100
Receive a text input for a calendar event
110
Resolve the text input to generate one or more suggested dates and times for the calendar event
120
Score the suggested dates and times to generate a ranked list of suggested dates and times for the calendar event
130
Provide the suggested dates and times based on the ranking
140

FIG. 1

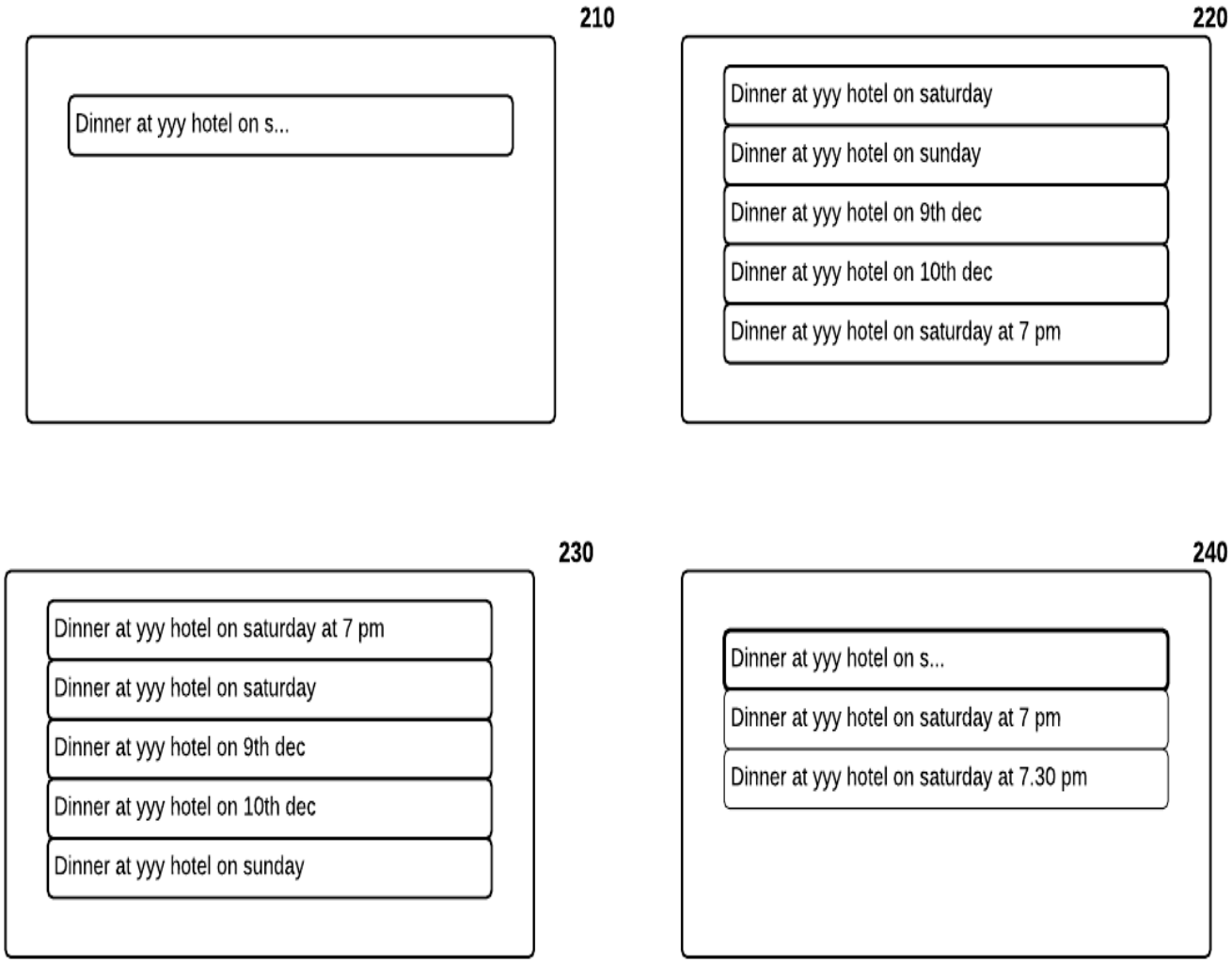
210
Dinner at yyy hotel on s...
220
Dinner at yyy hotel on saturday
Dinner at yyy hotel on sunday
Dinner at yyy hotel on 9th dec
Dinner at yyy hotel on 10th dec
Dinner at yyy hotel on saturday at 7 pm
230
Dinner at yyy hotel on saturday at 7 pm
Dinner at yyy hotel on saturday
Dinner at yyy hotel on 9th dec
Dinner at yyy hotel on 10th dec
Dinner at yyy hotel on sunday
240
Dinner at yyy hotel on s...
Dinner at yyy hotel on saturday at 7 pm
Dinner at yyy hotel on saturday at 7.30 pm

FIG. 2