March 26, 2018

Disambiguation of entity references in natural language interpretation

Steffen Meschkat

Disambiguation of entity references in natural language interpretation

ABSTRACT

This disclosure describes techniques for disambiguation of entity references in user input. The techniques can be implemented in computing systems that store data as identifiable entities along with relationships between the entities and are effective in user input with possible ambiguities. User input is scanned for references to entities generically. If an input name refers to more than one known entity known, and another entity name is detected in the same user input, the relationships between the entities are used to disambiguate the input.

KEYWORDS

- Disambiguation
- Natural language processing
- Conversational UI
- Voice input
- Entity reference

BACKGROUND

An important step in interpretation of natural language based user input is the detection of user references to known entities. This is relevant in the interpretation of textual input and particularly so in the case of voice input. Such references are commonly made by the mention of the names of entities.

For example, a voice input "Tell Jeff I'm late to our meeting" that references a person named "Jeff " (entity name: “Jeff”; entity type: “Person”) can be ambiguous, e.g., due to multiple occurrences of the name. One technique to address ambiguity is to impose a requirement to use more complete names in the input. For example, ambiguity can be reduced by using the complete name of the person. However, this approach does not resolve the ambiguity if multiple entities match the complete name.

Related entities can be utilized for disambiguation. In the provided example, if there is only one entity “Jeff” that belongs to the user’s workgroup, a reference to Jeff’s workgroup (e.g., Jeff from Project Red, where “Project Red” is a related entity) can enable disambiguation of the mention of “Jeff” in the voice input.

However, in current computer systems, user input is typically processed by one computer application at a time, while entities of different types are stored by different applications, e.g., an address book application that stores person names, a Project database application that stores workgroup names, etc.

DESCRIPTION

This disclosure describes techniques for disambiguation of entity references in natural language inputs to computers. The techniques utilize an operating system that stores and processes data in generic units or chunks, referred to as entities, similar to files in a traditional operating system. Computer applications are provided with entity references as inputs, analogous to how applications in traditional operating systems open files.

User intent to launch an application with a specific entity as input to the application is determined by interpretation of user input. User input is interpreted by the operating system and entity references are disambiguated with the knowledge of relationships between entities. The techniques are effective in interpreting user input to computing systems that store data by identifiable entities and relationships between the entities. The techniques are especially effective when related entities of different type, e.g., “persons” and “projects” are used.

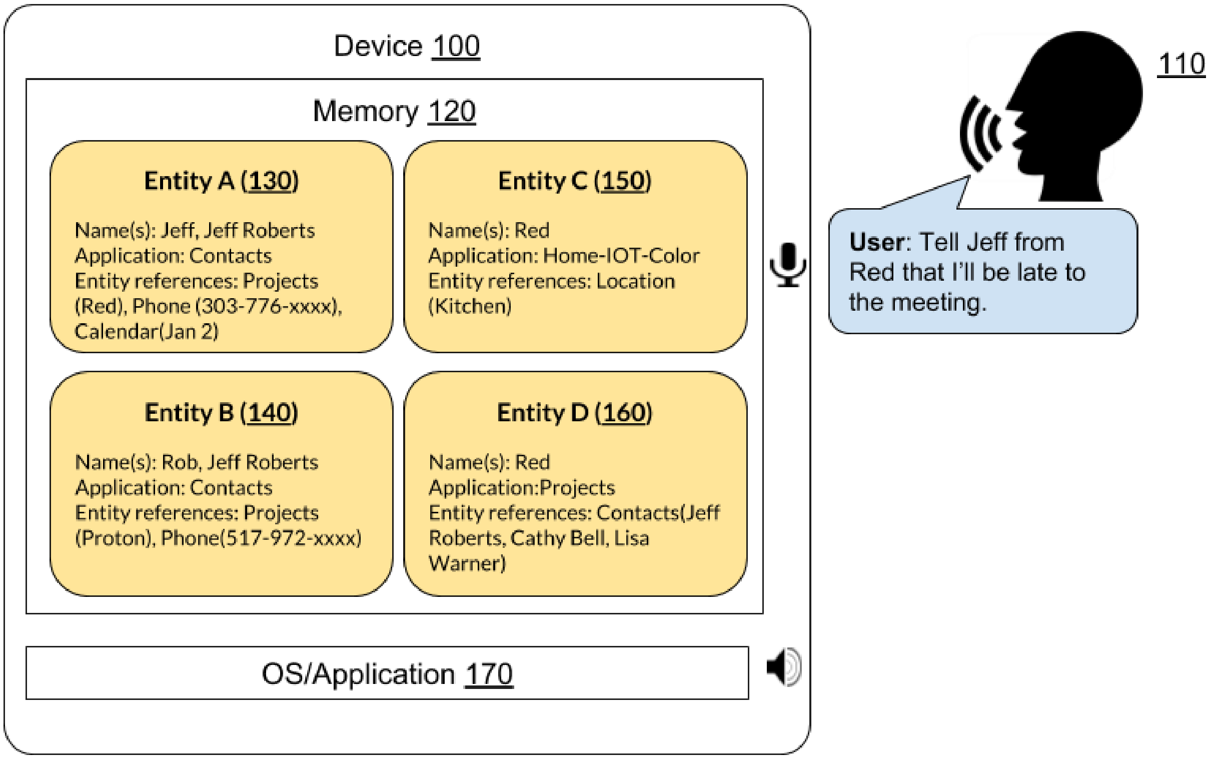
Device 100
Memory 120
Entity A (130)
Name(s): Jeff, Jeff Roberts
Application: Contacts
Entity references: Projects (Red), Phone (303-776-xxxx), Calendar(Jan 2)
Entity C (150)
Name(s): Red
Application: Home-IOT-Color
Entity references: Location (Kitchen)
Entity B (140)
Name(s): Rob, Jeff Roberts
Application: Contacts
Entity references: Projects (Proton), Phone(517-972-xxxx)
Entity D (160)
Name(s): Red
Application:Projects
Entity references: Contacts(Jeff Roberts, Cathy Bell, Lisa Warner)
OS/Application 170
110
User: Tell Jeff from Red that I'll be late to the meeting.

Fig. 1: Disambiguation of entity names using related entities

Fig. 1 illustrates a device (100) that implements an operating system or application (170) that utilizes techniques disclosed herein for disambiguation of entity names with knowledge of related entities. A user (110) provides voice input “Tell Jeff from Red that I’ll be late to the meeting.”

Entities A-D (130, 140, 150, and 160) are stored in memory (120). Each entity has one or more dedicated names including multiple names. For example, Entity A has “Jeff” and “Jeff Roberts” as names. The names can also include different names in different languages.

The entity names and relationships between entities are known generically to an operating system that implements the described techniques, in a manner similar to how a filename and the fact that two files reside in the same folder/directory, respectively, is known to a traditional operating system, even when the file content is not intelligible.

The operating system disambiguates an ambiguous reference, e.g., a reference to “Jeff” which is ambiguous since both Entity A and Entity B have the name “Jeff,” using the related entity reference "Red” in the user input, e.g., since there is only one “Jeff” related to a project “Red.” In response to the voice input "Tell Jeff from Red that I'll be late to the meeting," the system disambiguates both entities “Jeff” and “Red.” The voice input is then interpreted to be an instruction primarily about the person Jeff and not about the project Red. The reference to the entity “Red” is also disambiguated in the interpretation to refer to the project and not the color.

Each time an entity is created by an application and stored, the application can specify, one or more names to use to refer to that entity, and other related entities. References to related entities are specified by an internal data type (an entity reference), which is also provided to an application during initial entity creation. Different entities are referenced by distinct entity reference values and are stored separately.

With user permission, user input is scanned for references to entities generically by scanning for entity names since entities accessible by all applications are also accessible to the operating system. Occurrence of the name of a known entity in an input to the system is interpreted as a reference to the entity. If a name refers to more than one known entity, and another entity name is detected in the same user input, the relationships between the entities are used to disambiguate the references made by the names.

With increased usage of user assigned names in inputs to computers (e.g., in voice based inputs), there is the possibility of increased ambiguity in name references. For example, smart home devices such as speakers, smartphones, and other devices enable users to operate lights with a voice command such as "switch on the lights." However, since there may be multiple lights at the user’s location, such an ambiguous command may not be adequate for interpretation.

To disambiguate, lights can be named after the rooms they are in, e.g., "Kitchen lights," “living room lights,” etc.

Techniques disclosed herein facilitate disambiguation when the rooms are specified as entities and allow for increased flexibility in interpretation of user input. The interpretation is robust to variations in the order that the references are specified, e.g., “lights in the kitchen” and “kitchen lights” are both interpreted as referring to the same entity. Further, translations to other languages and to the use of synonyms in user input are supported.

The techniques are effective regardless of the specific entity types specified in the user input, since each stored entity has specific known names and relationships with other entities.

In situations in which certain implementations discussed herein may collect or use personal information about users (e.g., user data, information about a user’s social network, user's location and time at the location, user's biometric information, user's activities and demographic information), users are provided with one or more opportunities to control whether information is collected, whether the personal information is stored, whether the personal information is used, and how the information is collected about the user, stored and used. That is, the systems and methods discussed herein collect, store and/or use user personal information specifically upon receiving explicit authorization from the relevant users to do so.

For example, a user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature. Each user for which personal information is to be collected is presented with one or more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more such control options over a communication network. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. As one example, a user's identity may be treated so that no personally identifiable information can be determined. As another example, a user's geographic location may be generalized to a larger region so that the user's particular location cannot be determined.

CONCLUSION

This disclosure describes techniques for disambiguation of entity references in user input. The techniques can be implemented in computing systems that store data as identifiable entities along with relationships between the entities and are effective in user input with possible ambiguities. User input is scanned for references to entities generically. If an input name refers to more than one known entity known, and another entity name is detected in the same user input, the relationships between the entities are used to disambiguate the input.