March 26, 2018

Disambiguation of entity references using related entities

Steffen Meschkat

Disambiguation of entity references using related entities

ABSTRACT

This disclosure describes techniques for disambiguation of entity references in natural language processing. The disambiguation of ambiguous entity references is performed by seeking user confirmation of the intended reference. With user permission, the confirmation query or prompt utilizes phrases that include the name of the ambiguous entity and name(s) of related entities. Entities are stored along with names of related entities and designation of the nature of the relationship. The nature of relationship between stored entities is used to construct appropriate conjunctions in disambiguation phrases. The described techniques can be used in any natural language interface, e.g., text or voice, and can be implemented as part of an operating system, a virtual assistant application, etc.

KEYWORDS

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

BACKGROUND

In natural language based user interfaces, understanding user input involves system recognition of references to entities. The entity references occur by mention of the name of the entity in the input, e.g., text or voice. Since multiple entities can share the same name, such input can be ambiguous. Disambiguation techniques are utilized in such instances to enable the system to interpret the reference.

However, there can be situations in which the initial user input remains ambiguous, and additional information is solicited by the system from the user to disambiguate the user input. One way to solicit disambiguating information is to enumerate all possible interpretations of the reference and request the user to make a selection of the intended entity.

The construction of descriptions used for the multiple ambiguous entities has to unambiguous. For example, if the original user input is "Tell Jeff Roberts that I'll be late to the meeting," when there are two entities, e.g., persons with the name "Jeff Roberts" in an address book, simple disambiguation such as a question "Do you mean Jeff Roberts, or do you mean Jeff Roberts?" is insufficient to resolve the input.

In this instance, the system can provide additional information to overcome this problem. For example, with user permission, the system can access and provide additional information such the name of an affiliated organization or group for each contact, a birthdate, etc. Even with the use of this technique, the reference could still be ambiguous (e.g., since large organizations have many people with the same name, people share birthdays, etc.) Further, use of such information in a user interface may also be unsuitable in some instances, e.g., when the additional information is considered private or sensitive.

DESCRIPTION

This disclosure describes techniques for the disambiguation of entity references in natural language processing. The techniques can be utilized in any user interface that allows natural language input, e.g., conversation based user interfaces (UI) such as voice activated assistants, text based assistants, chatbots, etc.

Confirmation of ambiguous entity references is sought by the operating system by using the construction of phrases that includes the name of the ambiguous entity and name(s) of related

entities. Confirmation is made in the context of other known entities that can possibly be confused with each other. The system includes a suitable amount of contextual information to construct an unambiguous phrasing. Related entities are chosen that are most unique and are most likely to provide disambiguation.

When users provide consent for use of prior input data, the system utilizes previous user input history to select related entities that are the most likely. For example, if the user usually refers to people in the address book by their project affiliation, e.g., as indicated by the logs of previous user input, the system solicits disambiguation input accordingly by using project affiliation entities. Use of such prior input data or other user data is based on specific user permission for this purpose.

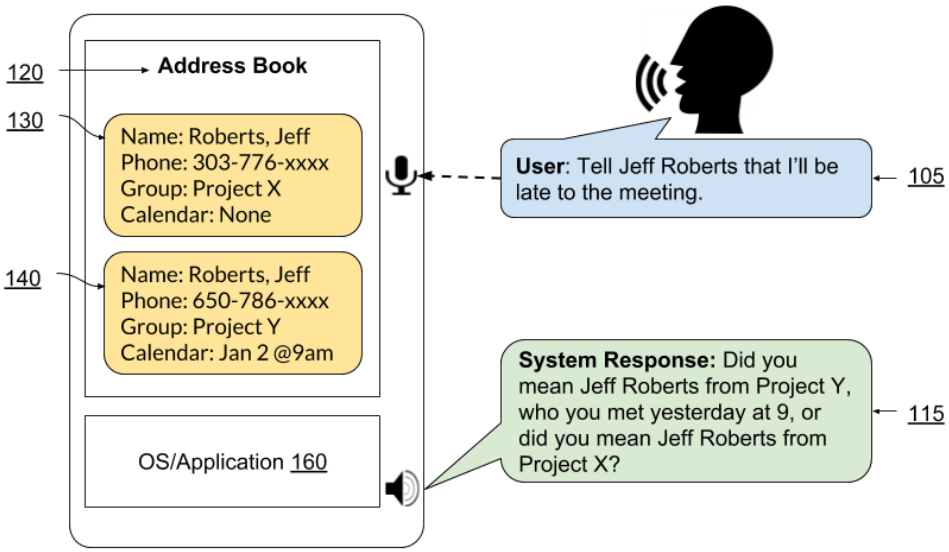

Fig. 1: Disambiguation using related entities

Fig. 1 illustrates an example interaction of a user with a voice assistant that utilizes disambiguation techniques described herein. The voice assistant accesses user data, e.g., address book, calendar, prior inputs, etc. upon specific user permission for use of such data for user interaction with the voice assistant. The user issues a command to the voice assistant to convey a

message (105), “Tell Jeff Roberts that I’ll be late to the meeting.” With user permission, the voice assistant, e.g., a standalone voice assistant application, an operating system, etc., accesses the address book (120) of the user to identify two entries of entities with the name “Jeff Roberts” (130, 140). The system further identifies related entities and determines that that both names have a respective workgroup name recorded in the address book, and that for one entity, there is also a calendar entry (140).

The system provides a response to the user (115) that includes a confirmation request to perform the disambiguation, e.g., "Did you mean Jeff Roberts from Project Y, who you met yesterday at 9, or did you mean Jeff Roberts from Project X?" A subsequent response from the user (not shown in Fig. 1) enables an identification of the original intended entity. While this example illustrates use of calendar and workgroup data for disambiguation, any user-permitted data can be utilized for this purpose.

While the foregoing example illustrates entity disambiguation with reference to entries in an address book, the techniques can be used for disambiguation of any type of entity. Computing devices that implement the described disambiguation techniques as part of an operating system store application data in generic units or chunks, referred to as “entities”. In this operating system, applications are launched with entities to operate upon, analogous to the manner in which applications are utilized to open files in traditional operating systems. User input is interpreted as the specification of an entity to be provided to an application.

In scenarios where the entity reference is recognized but ambiguous (multiple entities recognized by the system are consistent with the user input), unambiguous entity references are formed for the candidate entities, using user data permitted for use for such purpose. These references are presented for user selection of the intended reference. The unambiguous entity

references are formed from the name of the entities and the names of prominent or previously used related entities.

Entity storage is designed in a manner such that all entities are known to the operating system. In addition to the names of the entities, the system stores the names of related entities, as well as a designation of the nature of the relationship between the entities. The nature of the relationship is used to construct appropriate conjunctions in phrases constructed by the system that utilize the entity reference. For example, the relationship of a person to a home address can be expressed in an English phrase using the preposition "lives at."

Techniques disclosed herein can also be employed to confirm entity references that appear to be unambiguous. This is advantageous in scenarios where there may actually be an ambiguity, but one that is not recognized as an ambiguity, e.g., when other entities of that name are not recorded in the system. For example, even if only one "Jeff" is in a user's address book, the system can seek confirmation as to whether the user meant "Jeff Roberts on Project X."

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 natural language processing. The disambiguation of ambiguous entity references is performed by seeking user confirmation of the intended reference. With user permission, the confirmation query or prompt utilizes phrases that include the name of the ambiguous entity and name(s) of related entities. Entities are stored along with names of related entities and designation of the nature of the relationship. The nature of relationship between stored entities is used to construct appropriate conjunctions in disambiguation phrases. The described techniques can be used in any natural language interface, e.g., text or voice, and can be implemented as part of an operating system, a virtual assistant application, etc.