

# CLARA: a multifunctional virtual agent for conference support and touristic information

Luis Fernando D'Haro, Seokhwan Kim, Kheng Hui Yeo, Ridong Jiang,  
Andreea I. Niculescu, Rafael E. Banchs, Haizhou Li

Human Language Technology Department - Institute for Infocomm Research. 1 Fusionopolis way, #21-01 Connexis (South Tower), Singapore 138632

**Abstract** In this paper we present a multifunctional conversational agent which combines natural language search capabilities for two different domain applications: a conference information system and local tourist guide. The paper describes the corpora, architecture, algorithm and the mobile application created to interact with the users. Finally, some results obtained when using the proposed system in the context of an international scientific conference held in Singapore in September 2014 with more than 1200 assistants are provided.

## 1. Introduction

Nowadays, there is an increasing interest in using conversational agents for both web and mobile applications since they allow users to quickly find corporate or product information while, at the same time, engage users by providing them with relevant notifications about new products, offering recommendations, or simply by being able to handle user complains or feedback (i.e. chatbot capabilities).

There are several examples of conversational agents used for different domains in the literature, such as health-care [1], weather forecast [2], tutoring [3], tourism [4], etc. Probably, the most popular applications are Apple's Siri [5], Google Now<sup>1</sup>, and Cortana<sup>2</sup>. These systems are able to provide information for multiple tasks and domains including making appointments, sending text messages, providing weather or transportation information, or searching the web. However, up to the best of our knowledge, there is not any conversational agent specifically designed for conferences, which at the same time could handle local information about the place where the conference is held; which certainly is very useful for first time visitors.

The paper is organized as follows: in section 2, we describe the conference and tourist datasets used. Section 3 presents the system architecture explaining each module and algorithms for handling different types of searches. Lastly, in section 4, we show some system usage statistics collected when used in a real conference.

---

<sup>1</sup> <http://www.google.com/landing/now/>

<sup>2</sup> <http://www.windowsphone.com/en-sg/how-to/wp8/cortana/meet-cortana>

**Table 1.** Main statistics related to conference and tourism information domains

| Domain     | Concent  | Details |
|------------|--|---------|
| Conference | No. paper titles   | 633     |
| Conference | No. sessions   | 93      |
| Conference | No. keywords extracted from titles, abstracts and sessions names | 7519    |
| Conference | No. abbreviations and scientific terms                           | 1276    |
| Conference | No. authors  | 1695    |
| Conference | No. countries from the assistants                                | 46      |
| Tourism    | No. of different kind of restaurant-related places               | 28      |
| Tourism    | No. of places to search for restaurant-related information       | 120     |
| Tourism    | No. of different kind of food to search for                      | 81      |
| Tourism    | No. of main touristic attractions                                | 35      |

## 2. Data description

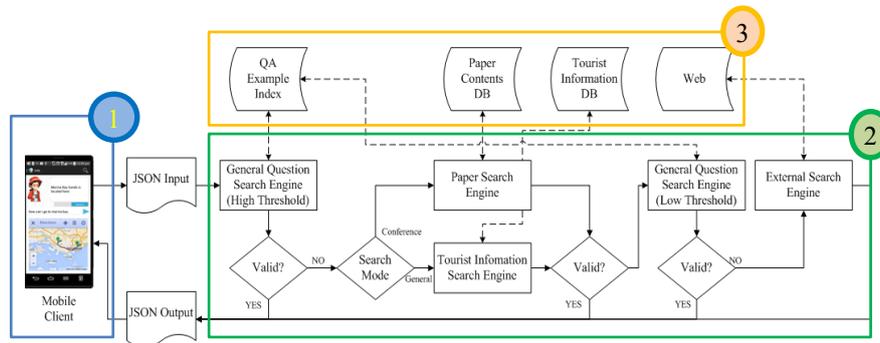
Since our goal was to deploy this application within the context of a real and large scientific conference, we decided to launch it during the 15<sup>th</sup> conference of the International Speech Communication Association (Interspeech), which is an annual conference where more than 1200 attendees from 46 different countries meet together to discuss and share information about speech-related technologies. The conference was held in Singapore from 14-18 of September 2014 and included 93 different sessions (see table 1).

The information concerning the conference was extracted from the official proceedings. In this case, we used only the titles, abstracts, and sessions but not the paper content itself. The main reason for not using the full text was to adjust better our system to the type of information people usually search for, i.e. by topic, nationality or affiliations, specific authors, or some hot-topic technology, rather than searching specific info about how an algorithm works or the final results of a system, which are detailed in the paper.

On the other hand, and considering that this conference was held in Singapore for the first time, we decided to provide some tourism information to visitors too. Taking into account that we have already deployed a tourist conversational agent [4] and a restaurant recommendation system [6], we decided to include these two functionalities into the new system, allowing the attendees to find useful information about sightseeing, transportation, shopping centers, food and beverage, as well as some general but important information about Singapore (see table 1).

## 3. Architecture description

The system architecture (Fig. 1) has three main components: 1) the client system implemented in a mobile application, 2) a websocket server which runs the



**Fig. 1** System Architecture

service and internally communicates with the orchestration and searching modules, and 3) the different resources (e.g. databases, dictionaries, and models) used to provide the information to the users and to enhance the system capabilities.

In more detail, the user poses a query using the graphical interface available on the mobile. With this information, the system creates a JSON message consisting of: the query, the domain (conference or tourist), and GPS coordinates, if the user allowed for sharing them. Then, the server calls the orchestration module which, depending on the specified domain, follows the following process:

First, the system searches in the index for generic questions or greetings like: ‘what can you do?’, ‘What is your name?’, or ‘how can I start using you?’ The search is done by retrieving the most similar examples in the index to the input query with a very high threshold to guarantee a high precision in the answer selection process. In case there is not an answer above the predefined threshold, the orchestration module calls either the conference search engine or the tourist search engine depending on the specific domain under consideration.

In the case of paper searches (conference domain), the system allows users to search by authors, affiliations, countries, titles, conference sessions and events, as well as general queries about conference facilities. Here, the system recognizes conference domain entities from the input by means of a fuzzy search algorithm; the algorithm is robust to a certain degree to input misspellings (this is particularly important for the case of authors’ names, where a high incidence of misspellings is expected). Then, the extracted entities are expanded with knowledge bases, including the information of keywords, abbreviations and synonyms. For each input, an SQL query is generated based on this semantic representation and it is used for searching papers, authors, or session information in the conference database.

For example, the system could capture the concept ‘Microsoft’ as an affiliation and ‘speaker recognition’ as a technical term from the user input ‘show me papers from MS on speaker recognition’, and then an SQL query with these constraints in the *where* clause would be prepared.

For the case of tourist information (tourism domain), the system is able to search for maps, restaurants, sightseeing locations, as well as for local and general information (history, currency, exchange rates, laws, etc). Here, the module includes a text classifier that is able to detect the topic of search. Depending on this classification, the system parses the query in order to provide meaningful answers.

For instance, in case the user is doing a search like ‘*show restaurants serving cheap Thai food in the nearby*’, the module extracts the following semantic information: (pricerange, *cheap*), (type\_of\_food, *Thai*) and (location, *nearby*). The same fuzzy matching algorithm as in the paper search engine is used for this parsing process. With this information, the system first disambiguates the location by replacing the word *nearby* with the GPS coordinates and searches for the district where the user is located. Next, the system creates a structured URL on a well-known restaurant website in Singapore, which will be used later to display the results on the mobile app. In addition, the system is also able to handle queries like: ‘*show me the nearest subway station*’ or ‘*a good shopping center in downtown*’. In these cases, the system creates maps or routes by using Google Maps API<sup>3</sup>.

Following with the algorithm, if there is still not an answer, the system attempts a new search in the index for generic queries, but using a more relaxed threshold. In the last instance, the system tries to search on internet by retrieving the result of a search using Wolfram Alpha search engine<sup>4</sup>.

Finally, the system sends back the answer to the mobile app by generating a JSON message containing the following information. a) The best agent to show the information (in case there is a change with respect to the original agent requested by the user, the mobile application switches to the new agent providing this way a feedback to the user regarding the agent used to answer the question). b) Feedback information displayed in a textbox near the avatar. c) Type of information to show: this informs the mobile if the information to display is a map, an external website, directly a HTML content, a list of papers/authors/sessions, or simply a chat answer. d) The URL to be displayed: this way it is possible to show maps, websites, pictures from sightseeing points, or restaurants information.

#### 4. Deployment and results

In order to implement the final user interface, a free mobile app, available at the Google Play and Apple’s App stores, was deployed in collaboration with researchers from the Quality and Usability Lab of Telekom Innovation Laboratories (TU Berlin). The full version of the app included several capabilities such as: a) basic search for papers, authors and sessions, b) conference schedule, c) the possibility of adding events to the user’s personal calendar, d) conference venue maps, e) direct access to the conference website, and f) a paper recommendation system.

In addition, the mobile application included a tab where the users could find our conversational agent (see Fig. 2). The agent screen was divided in three sections. 1) The avatar and a feedback textbox to display answers or summaries of the results. In addition, this section included a switch button for the user to select the type of information they were looking for. In case of looking for conference in-

---

<sup>3</sup> <https://developers.google.com/maps/>

<sup>4</sup> <http://www.wolframalpha.com/>



**Fig. 2** Aspect of the agent interface showing the three main areas. In the left, an example of search for tourist domain, in the right an example of search for papers.

formation, they could use the agent wearing a formal suit (conference domain system). Alternatively, in case of looking for touristic information, they should use the agent wearing the informal suit (tourism domain system). 2) The input box and submit button for sending queries to the system. 3) A multipurpose view area, where the information retrieved by the system is displayed (e.g. websites, list of clickable papers, maps, restaurants, history of previous searches and results, etc.). Finally, table 2 shows some of the usage statistics we collected during the conference. We can see that, in general, people used more the application for chatting with the system and searching for touristic information than for searching for conference information. The low usage of this latter information system can be probably due to the presence of the built-in search capabilities offered by the app, the recommender mechanism which also offered push notifications to the user, and some chat game sponsored by the conference organizers for people using the mobile app. On the other hand, we found that the number of queries the system could not answer was quite high (about 50% of the queries), but most of these (around 75%) corresponded to out-of-domain queries (i.e. mainly chat interactions), which the system could not detect as part of the chat and tried to get an answer from the index or external websites therefore being unable to process them.

**Table 2.** Usage statistics at the conference

| Concept  | Counts |
|--|--------|
| <b>No. total queries</b>                                 | 2360   |
| <b>No. different queries</b>                             | 1628   |
| <b>No. total different users</b>                         | 215    |
| <b>No. queries related with tourism/restaurants/maps</b> | 222    |
| <b>No. queries related with papers</b>                   | 480    |
| <b>No. chat interactions</b>                             | 1598   |

## 5. Conclusions and future work

In this paper we have presented a conversational agent that is able to answer to natural language questions formulated in the context of a scientific conference, as well as local information. In the conference domain, the system supports paper search by author names, topic, affiliation and country, while in the tourism domain the system is able to provide information related to touristic places, restaurants, and transportation. In addition, the system can chat with the users. Finally, the deployed system was tested during a large conference with satisfactory results. As future work we want to implement a mechanism to allow system administrators for conducting quick updates and sending notifications to the users. With this mechanism, if the system is not able to provide an answer, new information can be updated into the system and the user will receive a push notification on the mobile when an answer to his/her question is available.

## 6. Acknowledgments

We want to thank people from the Institute for Infocomm Research (I2R), Nanyang Technological University of Singapore (NTU), and the Insterspeech organizing committee for their support on deploying and testing the system. We also want to thank Dr. Jochen Walter Ehnes (I2R), Nicholas de Laczovich and Tilo Westermann from the Quality & Usability Lab of Telekom Innovation Laboratories (TU Berlin) for their work to integrate the virtual agent on the mobile app.

## 7. References

- [1] Beveridge, M. and Fox, J. Automatic generation of spoken dialogue from medical plans and ontologies. *Biomedical Informatics*, 2006 Oct;39(5):482-99
- [2] Maragoudakis, M. MeteoBayes: Effective plan recognition in a weather dialogue system. *IEEE Intelligent Systems*, 22(1), p. 66-77.
- [3] Johnson, W. L., Valente, A. Tactical language and culture training systems: Using Artificial Intelligence to teach foreign languages and cultures. *Proc. IAAI*, (pp. 1632-1639), 2008.
- [4] Niculescu, A. I., Jiang, R., Kim, S., Yeo, K-H., D'Haro, L. F., Niswar, A., Banchs, R. E. SARA: Singapore's Automated Responsive Assistant, a multimodal dialogue system for touristic information. *Proceedings of 11th International Conference, MobiWIS 2014, Barcelona, Spain, August 27-29, 2014*, pp. 153-164.
- [5] Bellegarda, J. Spoken Language Understanding for Natural Interaction: The Siri Experience. In *Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice*, p 3-14. 2014.
- [6] Kim, S., Banchs, R. E. R-cube: a dialogue agent for Restaurant Recommendation and Reservation, *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Special Session on Chatbots and Dialogue Agents, Siem Reap, December 9-12, 2014*.