Institute of Informatics
Logics and Security Studies

Technical Report on Development of Virtual Chat Agent

Samira Shaikh, Ken Stahl
Date – September 16, 2009

ILS Tech Report: [003]

Available for download from:
http://www.ils.albany.edu

Brief Statement of Work

The purpose of this research is to advance the understanding of the behavior of small groups and to develop automated capability to monitor, probe and then monitor responses to probes in an online chat room or other discussion forums.

This document in structured as follows – in Section 1, we briefly discuss the phase 1 of our project development cycle, namely, data collection from internet chat; in Section 2, we outline the virtual chat agent (VCA) system engineered to automatically respond to utterances in internet chat rooms; in Section 3, we provide technical details about the VCA system.

This is an ongoing project, and this report will be updated from time to time to reflect emerging changes.

1. Data Collection

We collected chat room interactions of small groups of participants using in house chat software. Participants were recruited from people affiliated with the University at Albany and included students, alumni and faculty members. In our pool of 12 participants, backgrounds and ages were also varied. We collected a total of ~20 hours of chat interactions spread out in 14 individual sessions. In each session, a group of 4-5 participants from our pool would log on and chat with each other for 90 minutes on a variety of topics. In some sessions, they were given specific tasks to perform – such as selection of a candidate to fill a job vacancy, and also some of the participants would play a designated role – such as discussion leader. We have a total 7317 chat utterances in our corpus.

Extended report on Data Collection for this project, forthcoming.

2. Virtual Chat Agent Development

2.1 Rationale

The idea is to exploit the dialogue mechanism underlying HITIQA to drive the dialogue in VCA. HITIQA does not support the task-oriented dialogue in the traditional sense (i.e., fill in elements of a plan or a template). Instead, it more opportunistically explores attributes and aspects of a topic under consideration. In HITIQA the topic is defined by the information contained in the user's question. This information (which may be quite limited) is used to mine outside data sources (e.g., a corpus, the web) in order to locate/learn additional information about this topic. The objective is to identify some of the salient concepts that appear associated with the topic but are not directly mentioned in

the question. Such associations maybe postulated because additional concepts are repeatedly found near the concepts mentioned in the query (here: near = in the close linguistic context, e.g. same passage).
Unlike in HITIQA, where the topic is explicitly (though only partly) defined by the user's question, a chat may consist of a series of topics that change with little overt indication and may additionally overlap in time and space. Each new utterance may introduce a new topic; however, conversational norms place some restrictions on when and how topic changes may occur.

Analogously to HITIQA queries, chat utterances will be used to develop topic frames by mining open data sources for additional material and the extracting attributes as before. Each subsequent utterance will either address one of the identified aspects or it would introduce new ones (or new clues to obtain them). In addition, we may attempt to model each participant position towards the topic.

We are currently designing a prototype that we will demonstrate in November of 2009. This prototype will be inserted in a chat room where it will attempt to

1. Provide additional information/concepts w.r.t the topic under discussion
2. Introduce a new topic on the basis of the topic under discussion

without being detected as a non-human participant. VCA engagement level in this demo will be quite minimal.

2.2 VCA Mode of Operation

We present below a typical mode of operation for the VCA.

1. Every utterance in the chat by any user becomes a candidate for a response by the bot. We employ certain filters to select utterances that qualify for further processing. For every utterance, we
    - remove all punctuation
    - remove all emoticons
    - remove all stopwords
    - remove any user nick names

    If there are any words left after the filtering, we consider that chat utterance to be a candidate for a response.

    For example, let the utterance be – I really like sushi too, alex. ☺
    which becomes – like sushi
    and becomes a candidate for response.

2. If an utterance is selected to be responded to by the agent, we remove the emoticons and user nick names from that utterance and send it to Identifinder and Stanford POS tagger.
    For eg. – I really like sushi too, alex. ☺
    becomes – I really like sushi too.
    We use the results from named entity tagging and pos-tagging to build a data frame.

3. In parallel, we send the utterance (after removing emoticons and user nick names) to Google AJAX api and retrieve the first eight documents returned by Google. Or in the offline mode of operation, the query is sent to InQuery.

4. We segment the documents retrieved into paragraphs and cluster them using HITIQA's clustering method.

5. We build an utterance frame from the chat utterance. In a corresponding step we build data frames from the seed paragraphs in the clusters. Using a scoring method we rank the data frames based on their closeness with the utterance frames.

6. We will select the most appropriate entity from the most closely matched data frame to create a response.

2.3 Future Directions

We are currently working to improve the scoring algorithm so that the most informative and appropriate response may be built from the most closely matched data frame.

3. Technical Specification of VCA

Below are the system and software requirements to run the VCA.

Run time environment – Redhat Linux 4
Software -

- OpenFire Spark chat client - XMPP chat
- Smack API to develop custom chat client – VCA
- MySQL server to log chat activity
- Spark chat client runs on all major platforms.
- Google AJAX API to retrieve documents from the web
- InQuery Indexing Engine – for 'offline' mode of operation
- IdentiFinder for NE tagging + Wine to run IdentiFinder on Linux
- Stanford POS tagger
- RiTa toolkit for Wordnet
- HITIQA v.0600