Introducing Ratatouille: a Generalizable Goal-Oriented Dialog Bot

Team M. Amram – J. Toledano
Faculty N. G. des Mesnards – T. Zaman
Company L. Gerdes – R. Sehgal – I. Pyzow

Problem Statement

Commercial solutions use **human** workforce to frame dialog with **rules**

Domain knowledge
- **Expertise** required to formulate business use case

Conversation transcripts
- Sample **dialogs** required to scope bot features

Business analysts
- Formulate a **base dialog flow** for a given use case
- **Handcraft** a specific series of rules from base dialog flows

Rule-based dialog flow
- Bot leads conversation using preset **question-based flow**
- Bot **classifies** user responses using its handcrafted rules

Our solution leverages **deep learning** to improve **generalizability**

Structured knowledge
- **Database** of structured information required to answer user requests

Extensive conversational data
- Thousands of **labeled conversation** transcripts required to use deep learning

Deep architecture
- Deep learning algorithms **infer patterns** from textual data to frame any dialog

Generalizable model

Can be **extended** by:
- **Switching** database
- **Incorporating** new features by **generating** new conversations
- **Curating** transcripts for any business **use case**

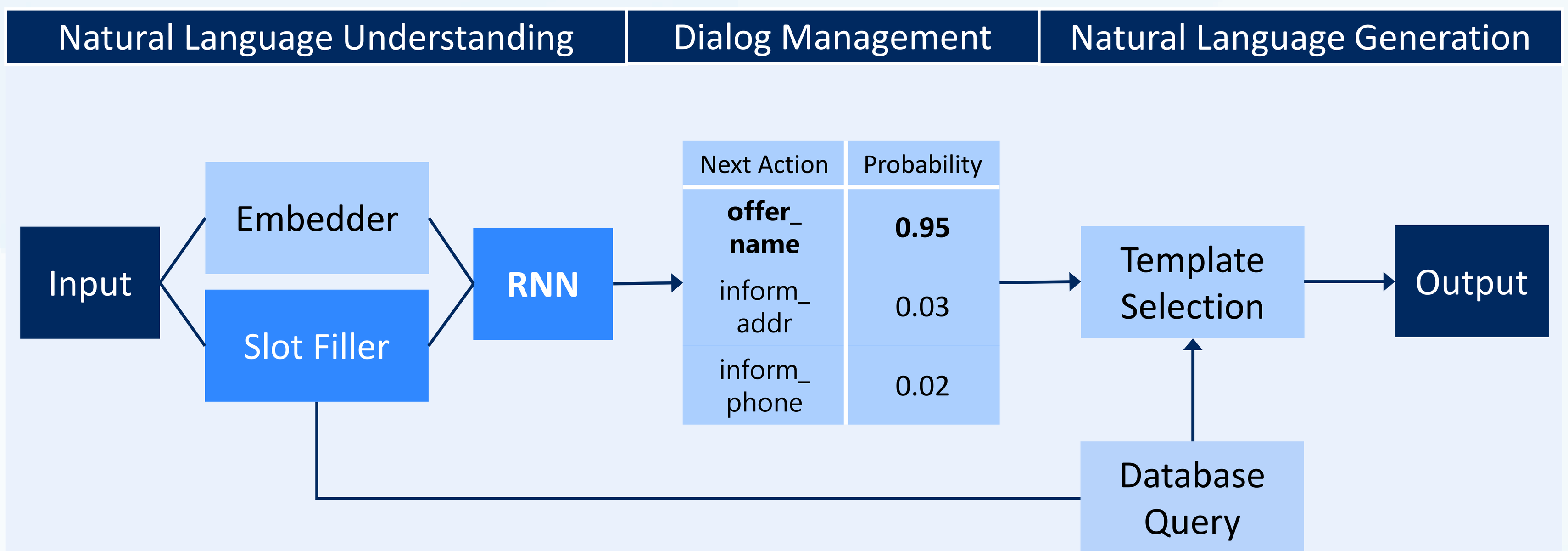
Data Integration & Architecture

Two **enhanced sources** fuel the restaurant recommendation task

Structured Database
- Information about 1,000 restaurants in **Boston**, **Cambridge**, and **Waltham**
- Data **collected** using APIs from Yelp, Zomato, and OpenTable
- Set of scripts **automates** data integration and cleaning

Transcripts
- More than 3,000 **open-source** conversation transcripts published by University of Cambridge
- Augmented with **new features** and **automatically generated sentences** by bespoke parsers

Our **end-to-end** architecture **predicts** the bot's next response



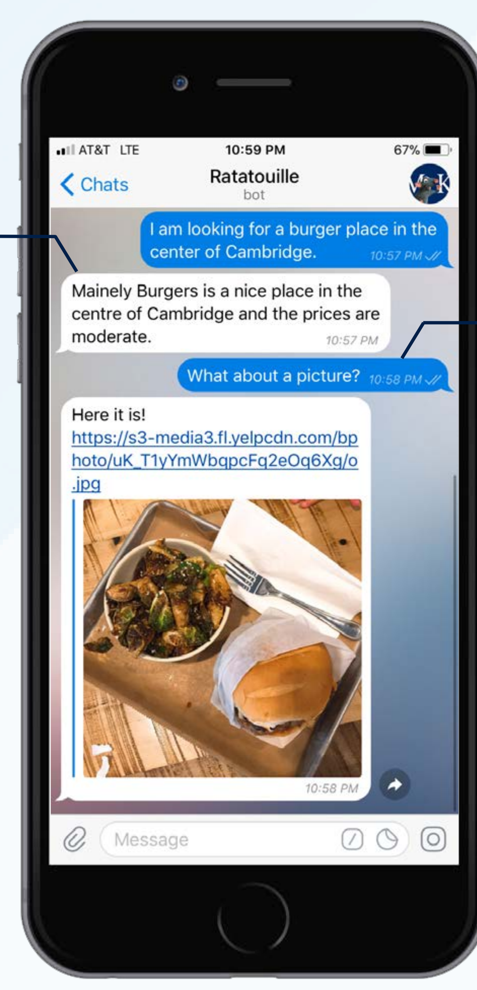
Demonstration Application

Bot takes into account **food type constraint** to suggest a restaurant

Phone number can be retrieved and clicked on

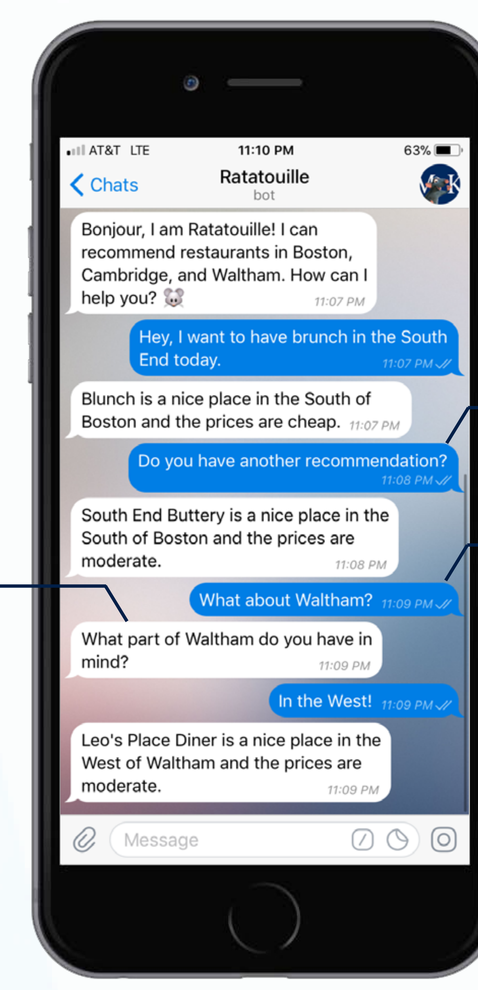
More details, such as the **address**, can be requested

Bot deals with **food type** and **neighborhood constraints**



A **picture** can be requested by the user

Bot understands city switch and inquires about **new preferred area**



Alternatives can be asked for

User can **switch between cities** within a conversation

Project Timeline

On-campus research				On-site internship		
February General literature review	**March** End-to-end architectures	**April** Building Informative DB	**May** Implementing Bot modules	**June** Release of Alpha version	**July** Example level generalizability	**August** Feature level generalizability

Impact

	Customer acquisition	Churn reduction	Cost reduction
Vertical	• Display **advanced capabilities** to prospective customers • Meet customer expectations • **Adapt** rapidly to new customer use cases	• Act on customer **preferences** • Automate customer **satisfaction** analysis • Answer questions with high accuracy **24/7**	• Automate repetitive tasks • Allow exceptional people to **focus** on high-value problem solving • Scale up and down depending on customer requirements
Examples	• **User-friendly** solutions bring about massive adoption	• **Brands** use bots to retain tech-savvy customers	• **Large-scale** implementations have a proven track record for generating **value**

Path Forward

New Use Case

Methodology to apply the architecture to a new business use case:
- Formulate the business use case as **recommendation task**
- Gather and **curate** thousands of conversation transcripts
- Build the corresponding informative database by **scraping** the web
- **Train** the core deep learning modules

Algorithm

Promising research-stage architectural developments:
- **Memory Networks:** RNN that selects and stores relevant dialog chunks in memory
- **Frames Tracking:** adding a memory module to rewind the dialog
- **Reinforcement Learning:** takes into account the future turns of the conversation to optimize the local dialog state

Infrastructure

From a prototype to production-ready solution:
- Training the core RNN with **GPU** reduces training time from 7 hours to 30 minutes
- **Cloud hosting** allows the bot to communicate with several users simultaneously to improve scalability