Helping Non-Experts Build Speech-Enabled Online CALL Courses

Manny Rayner

(joint work with Claudia Baur, Pierrette Bouillon, Nikos Tsourakis)
Overview

- CALL-SLT → Open CALL-SLT
- Constructing CALL-SLT courses
- Deploying CALL-SLT courses
- And next…
Overview

- CALL-SLT → Open CALL-SLT
  - Constructing CALL-SLT courses
  - Deploying CALL-SLT courses
  - And next…
A brief history of CALL-SLT

- “Spoken translation game” project
  - Funded by Swiss NSF, started mid-2009
- Second phase started 2014
Original idea

- We’d built this cool speech translation app
- People who played with it seemed to pick up spoken language skills
- Can we turn it into a spoken CALL app?
Project proposal

- Hm… MIT already did something similar
- We need to be different (research, right?)
- Linguistically motivated, grammar-based recognition!
- Regulus platform!
Project proposal

- Translate from L2 to human-readable interlingua based on L1
  - Prompts in interlingua
- Speech translation from L2 to L1
- Match result against prompt
First implemented version

- Deployed on desktop machine
- Typical prompt:
  « DEMANDER DE_MANIÈRE_POLIE TABLE 2 PERSONNE(S) »
- Typical spoken responses:
  - “I would like a table for two”
  - “A table for two people please”
Second implemented version

- Hm… no one can install system at home
  - Redeploy on internet, so actually usable
- Redesign interlingua
  - Make prompts more natural
- Content still unmotivating
  - Students not interested
Add scripts to allow interactive dialogues
Ad hoc rewriting rules to make prompts more natural
Original elegance of design gone
Only experts can implement content
Only experts can deploy the system
Evaluation version

- … but students do now want to use it
- Eight interactive English lessons for German 12-year-olds based on textbook
Evaluation in Swiss schools

- Q4 2014, German-speaking Switzerland
  - 15 schools
  - ~225 students
  - ~25K logged utterances
- Teachers and students both liked it

(Mostly work by Claudia Baur: several papers, 2013-2015)
Bottom line so far

- We can build useful courses, but…
- … the framework is ugly
- … it takes too long
- … only experts can do it
Can we do better?

BASIC DESIGN PRINCIPLES

- “Build One to Throw Away”
- “Keep It Simple, Stupid”
- “Web 2.0”
Build one to throw away

- Do we need the sophisticated recognition architecture?
Do we need the sophisticated recognition architecture?

- No
Recognition

- Recognition task is highly constrained
  - Prompt defines responses
  - Sophisticated recognition unnecessary
- Use enumerative approach
  - First cut: just list possible responses
  - Add simple grammar if necessary
Keep It Simple, Stupid

- Minimal formalism
- Less is more
- Do not require specialist knowledge
Web 2.0

- “Every downloader is a potential uploader”
- Be inclusive
- Paradigm: HTML
Overview

- CALL-SLT → Open CALL-SLT
  - Constructing CALL-SLT courses
- Deploying CALL-SLT courses
- And next…
Different kinds of designers

- Six different levels of complexity

- Lowest level: basic web-literate
  - Writing, saving and uploading files

- Highest level: non-trivial software skills
  - Ability to debug a simple program
Six levels

1. Prompts and responses
2. Multimedia
3. Regular expressions and templates
4. Scripts
5. Gamification
6. Advanced scripts
Level 1
Prompts and responses

- Simplest kind of course
- List of prompt/response pairs
- Prompt is piece of text
- One or more responses
Level 1
Prompts and responses

Prompt
Lesson pronunciation_h
Group 4
Text/french Dis que: tu déteste les légumes
Response i hate vegetables
Response * i ate vegetables
EndPrompt
Level 1
Prompts and responses

dis_que : tu détestes les légumes

I ate vegetables
Level 2 Multimedia

- Same as Level 1
- ... except that Prompts can have multimedia
Level 2
Multimedia

Prompt
Lesson
Multimedia  who
Text/english  pikachu.jpeg
Response  Who is it?
pikachu
Response  it is pikachu
Response  it’s pikachu
EndPrompt
Level 2
Multimedia

CALL-SLT
A PROTOTYPE CONVERSATION PARTNER

That is pikachu
Level 3
RegExs, templates, grammar

- You often have
  - Similar responses in the same Prompt
  - Similar Prompt units

- Level 3 gives tools for handling this
Level 3
RegExs, templates, grammar

i (want | need) a single room please

instead of

i want a single room
i need a single room
i want a single room please
i need a single room please
Level 4 Scripts

- Link up prompts to create a dialogue
- Simple XML-based scripting language
- Script = set of `<step>` units
<step>
  <id>ask_for_number_nights</id>
  <group>room_for_number_of_nights</group>
  <limit>is_one_night_okay</limit>
  <success>ask_type_of_room</success>
</step>
Level 5 Gamification

- Gamification may increase motivation
- Simple score/badge framework
- Designer can specify badges, thresholds, penalties, bonuses etc
Level 5
Gamification
Level 6
Advanced scripts

- Allow multiple paths through script
  - “Conditional steps”

- Make one step depend on another
  - “Semantic tags”
Level 6
Advanced scripts

<step>
  <id>enjoy_your_stay</id>
  <group>thanks</group>
  <limit>exit</limit>
  <success cond="level >= silver">
    is_everything_okay
  </success>
</success>
<success>exit</success>
Overview

- CALL-SLT \rightarrow \text{Open CALL-SLT}
- Constructing CALL-SLT courses
  - Deploying CALL-SLT courses
- And next…
Uploading courses

- In principle, very simple: copy course to server and run it
- In practice, less straightforward
  - Many users sharing same server
  - Stop users interfering with each other
  - Hide complexity of upload process
Uploading courses:
file structure

- Each user owns a namespace directory
- Namespace directory contains one or more course directories
- Course directory has up to 4 subdirectories
  - course description
  - multimedia
  - scripts
  - doc
Uploading courses: what the user sees

Five step process
- UPLOAD
- SELECT
- COMPILE
- TEST
- DEPLOY
Uploading courses: UPLOAD

- What the user sees
  - Connect to server
  - Drag and drop directory
Uploading courses: UPLOAD
Uploading courses: SELECT

- What the user sees
  - Select course from menu
  - Confirmation
Uploading courses: SELECT

- What the user sees
  - Select course from menu
  - Confirmation

- What happens on server
  - Check FTP directory
  - Register new courses
  - Copy to compilation directory
Uploading courses: SELECT
Uploading courses: SELECT

CALL-SLT
A PROTOTYPE CONVERSATION PARTNER

Step 1: Select a course
- english_course
- hello
- pronunciation
- toy
- visual

Step 2: Compile

Step 3: Test

Step 4: Release

Namespace:
Course:
L1: L2:
Uploading courses: SELECT
Uploading courses: COMPILER

- What the user sees
  - Press Compile button
  - Receive feedback
Uploading courses: COMPILE

What the user sees
- Press Compile button
- Receive feedback

What happens on server
- Compiles course description
  - Creates Nuance grammar, tables etc
- Compiles Nuance grammar
- If successful, adds touchfile
Uploading courses: COMPILÉ
Uploading courses: COMPILER

CALL-SLT
A PROTOTYPE CONVERSATION PARTNER

Tutorial: Writing CALL-SLT Lite courses

Step 1: Select a course  Step 2: Compile  Step 3: Test  Step 4: Release

- 2 templates internalized,
- 4 template applications expanded,
- Removed 38 incorrect responses duplicating correct ones,
- Found 45 units and 227 responses for namespace "Visual",
- Stored 6 scripts,
- Loaded 0 multimedia file definitions,
- Loaded 0 phrase scores,
- All groups in scripts are defined,
- All multimedia references in prompts are defined,
- Not yet checking multimedia files,
- No Lite compile errors in namespaces "visual".

Compile:
Uploading courses: TEST

- What the user sees
  - Press Test button
  - Receive feedback
Uploading courses: TEST

What the user sees
- Press Test button
- Receive feedback

What happens on server
- Copies course dir to staging server
- Recompiles all course descriptions
- Redeploys system with new Nuance grammar
Uploading courses: TEST
Uploading courses: TEST
Uploading courses: RELEASE

- What the user sees
  - Press Release button
  - Receive feedback
Uploading courses: RELEASE

- What the user sees
  - Press Release button
  - Receive feedback

- What happens on server
  - Copies course dir to production server
  - Recompiles all course descriptions
  - Redeploys system with new Nuance grammar
Uploading courses: RELEASE
Uploading courses: RELEASE
Overview

- CALL-SLT → Open CALL-SLT
- Constructing CALL-SLT courses
- Deploying CALL-SLT courses
  - And next…
So what now?

- Just about to begin alpha testing
- External users starting next week
- First courses
  - Practice academic terminology
  - Shopping in Geneva
  - Telephone conversation
So what now?

- What courses can users build?
- Can they be deployed on an interesting scale?
- We should know more soon
To learn more…

http://callslt.unige.ch/demos-and-resources/
Thank you!