**Content**

1. INTRODUCTION 3

1.1 Project Goal 3

1.2 Background knowledge 3

1.2 Concepts generation (Classes & Instances) 4

1.3 Intent generation 8

2. KEY FUNCTIONS 9

3. SOFTWARE ARCHITECTURE 11

3.1 Program List 15

3.2 API Interface 16

Terminology

* **Concepts** – classes and Instances.in the ontology.
* **Intention**( Every sentence can be interpreted into only one Intention and several semantic parameters)
* **Zenbo API** (we can use DS service via Zenbo API)
* **Dialogue System**(Includingmany subsystems .Main propose is to generate next sentence)
* **CSR**(Continuous Speech Recognition: Including acoustic model and language models, we use Google’s online API and Nuance offline solution)
* **STT**(Speech To Text)
* **Spoken Language Understanding** (got result from CSR and translate every sentence into only one Intention/appid and several semantic parameters)

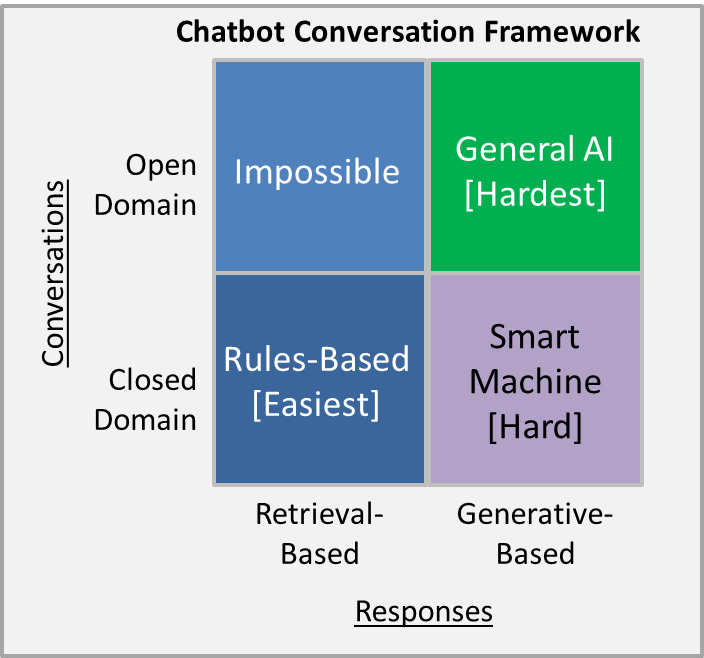
Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Alex Chuang | 2017/03/01 | Initial draft | 0.1 |
| Alex Chuang | 2016/04/02 | Add more description for the listed figures | 0.2 |

# INTRODUCTION

## Project Goal

This project uses retrieved-based question answering engine to help improve customer service experience. By automating answers to common questions via our dialogue system, we're able to improve the online shop experience by providing real-time 24/7 support, while also manage operating costs through efficiency.

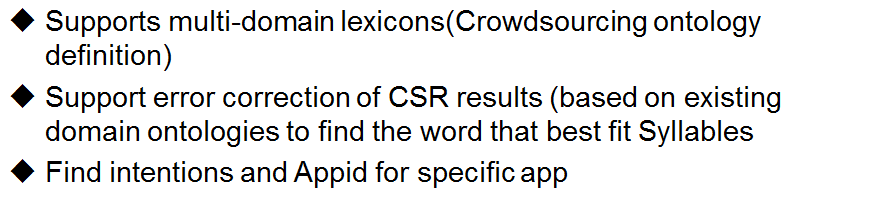


## 1.2 Background knowledge

* **Dialogue System**

Dialogue system gets inputs from CSR module and provides response to client App via Zenbo API. (API exported for app to use)

* **Semantic language Understanding**



Spoken language understanding helps dialogue system choose the most appropriate response for an utterance after understanding user’s meaning. For n-best speech recognition result, it retrieves the most likely intentions and extracts related information based on current dialogue status and the dialogue history for a specific user. Dialogue manager can then decide which module to handle current utterance based on the result.

* **IRP editor**

Web editor used to define intents/plans/concepts of apps.

Please check following link to define them.

DDE editor system:

<https://stage-developer.asus.com/tools/ds-editor.jsp>

* **Intent**

In the rum time, for an utterance which is recognized by speech recognition module, SLU outputs the best intentions and the related information from the knowledge database. SYSTEM ARCHITECTURE

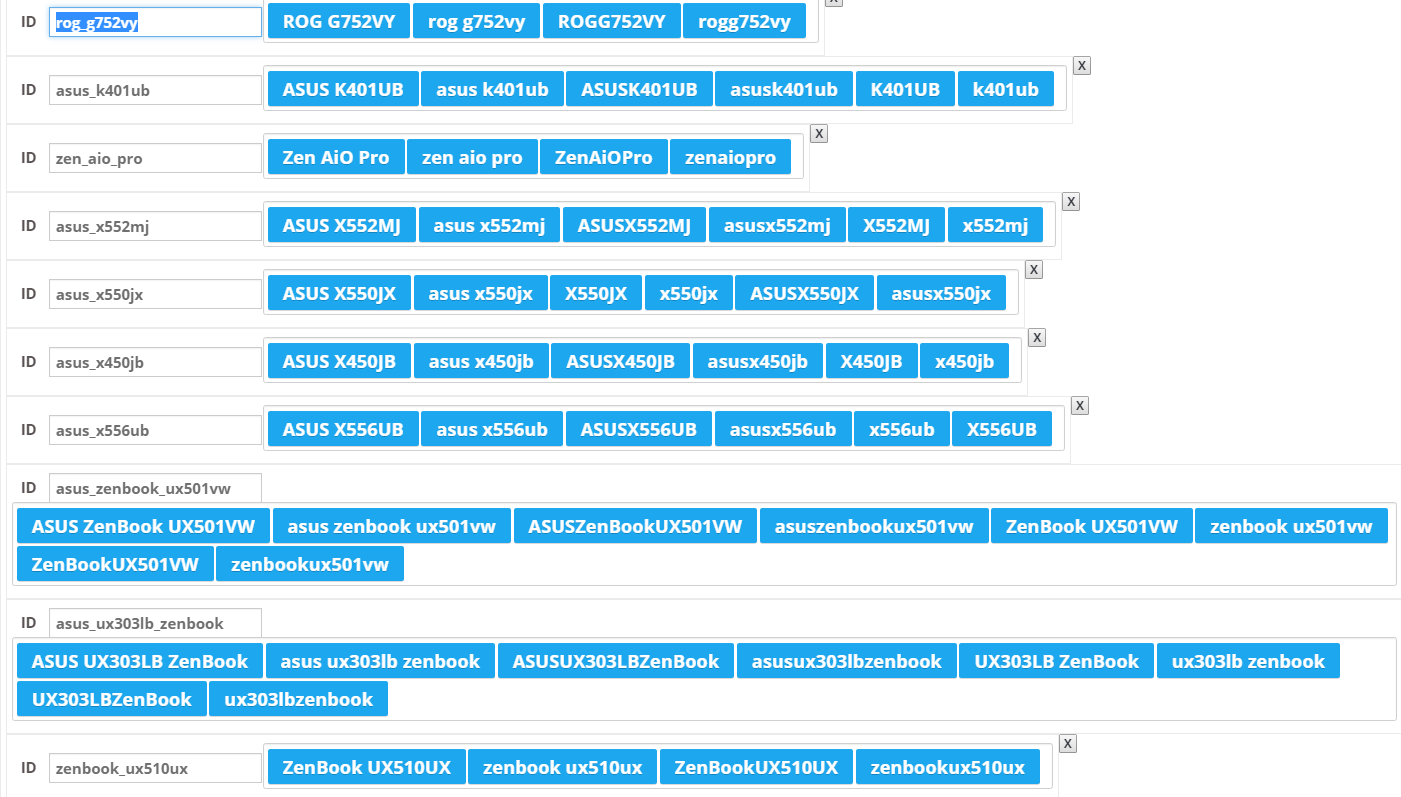
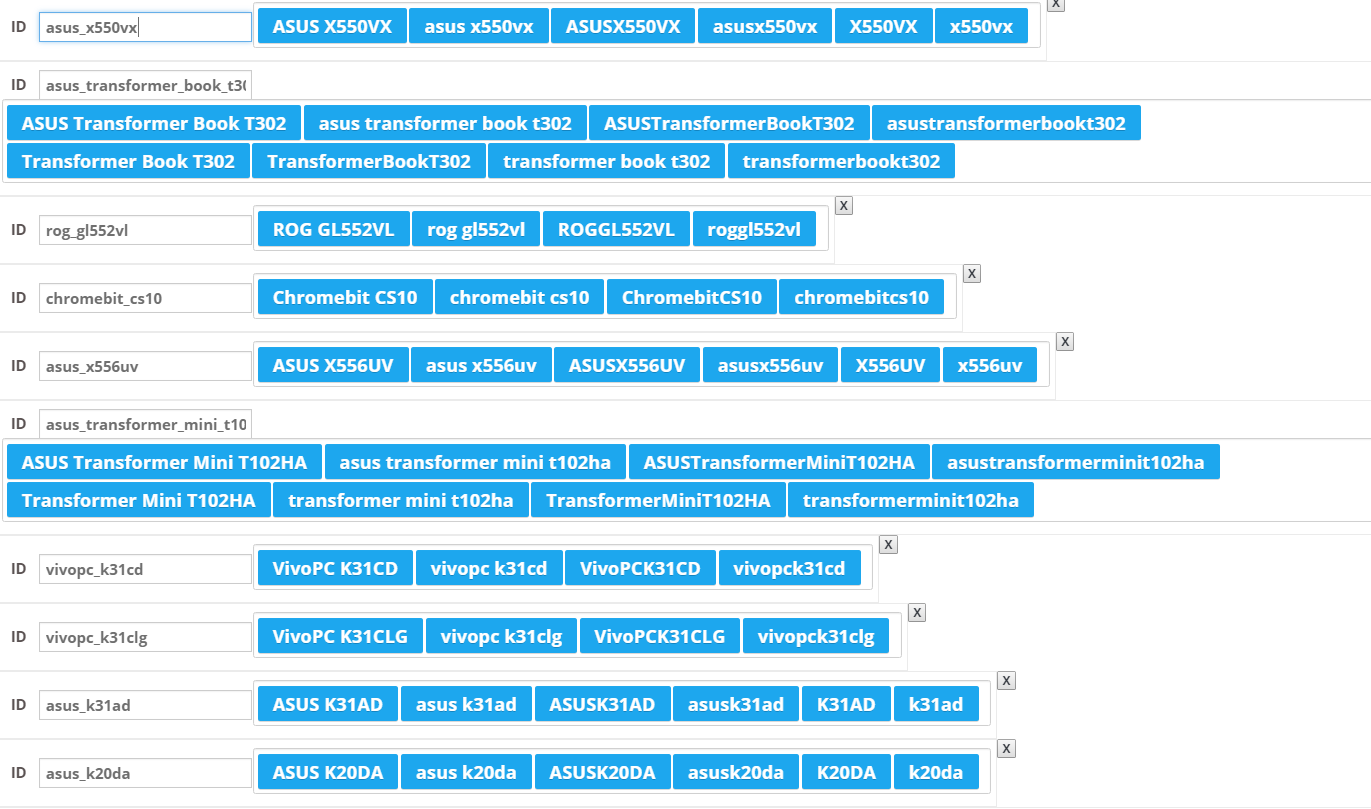
## Concepts generation (Classes & Instances)

From observation, we define six main intent types and several sub intent types used to define intents.

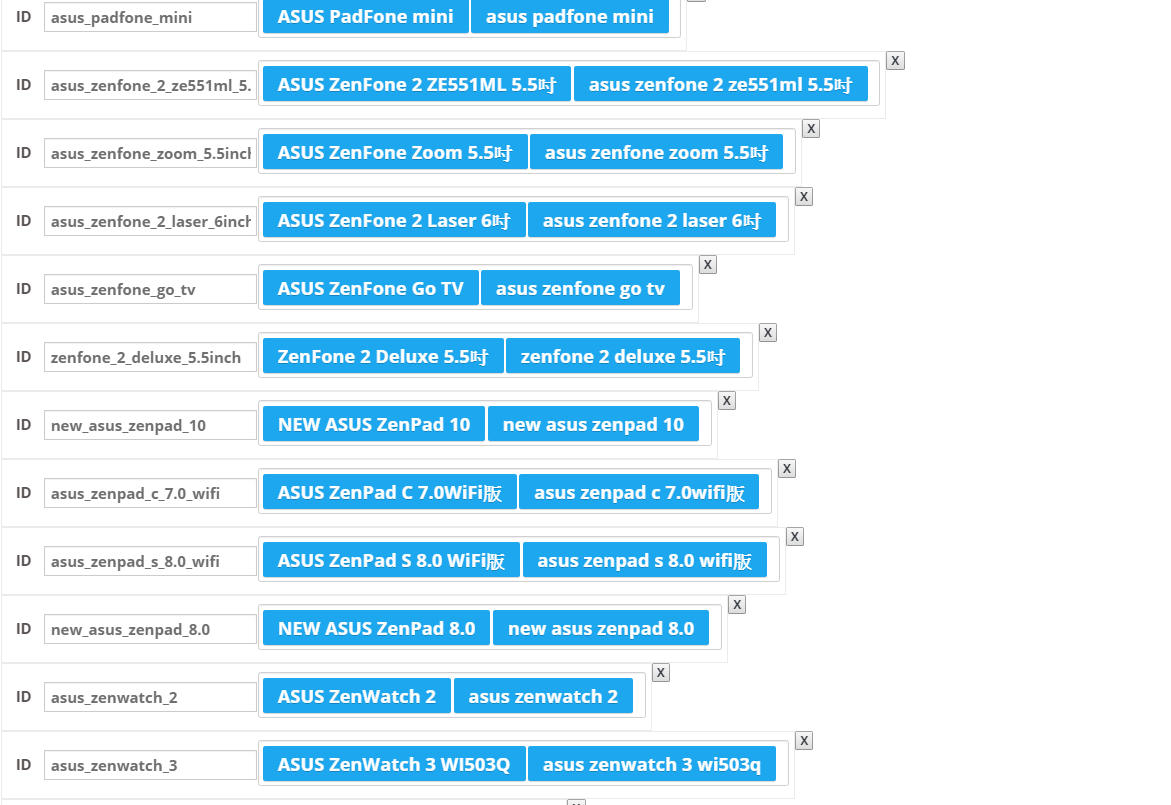
Class/instances of the ontology

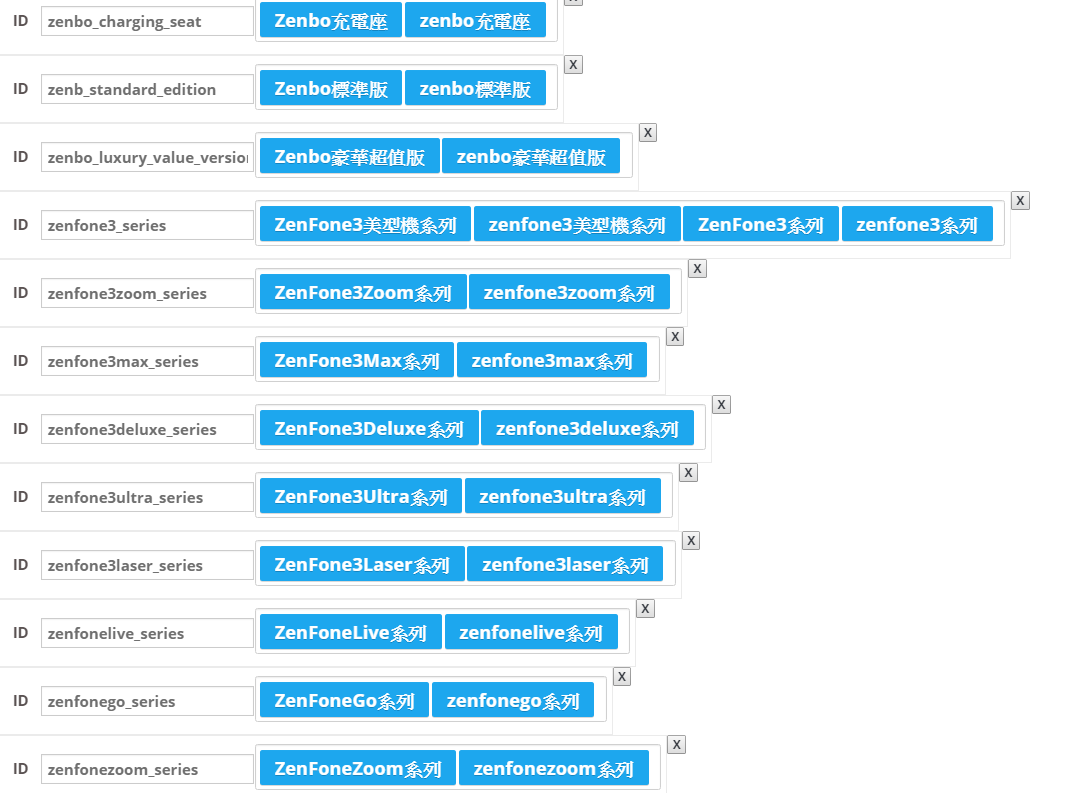
* **Product**
* **HowToKnow**
* **ApplicationObject**

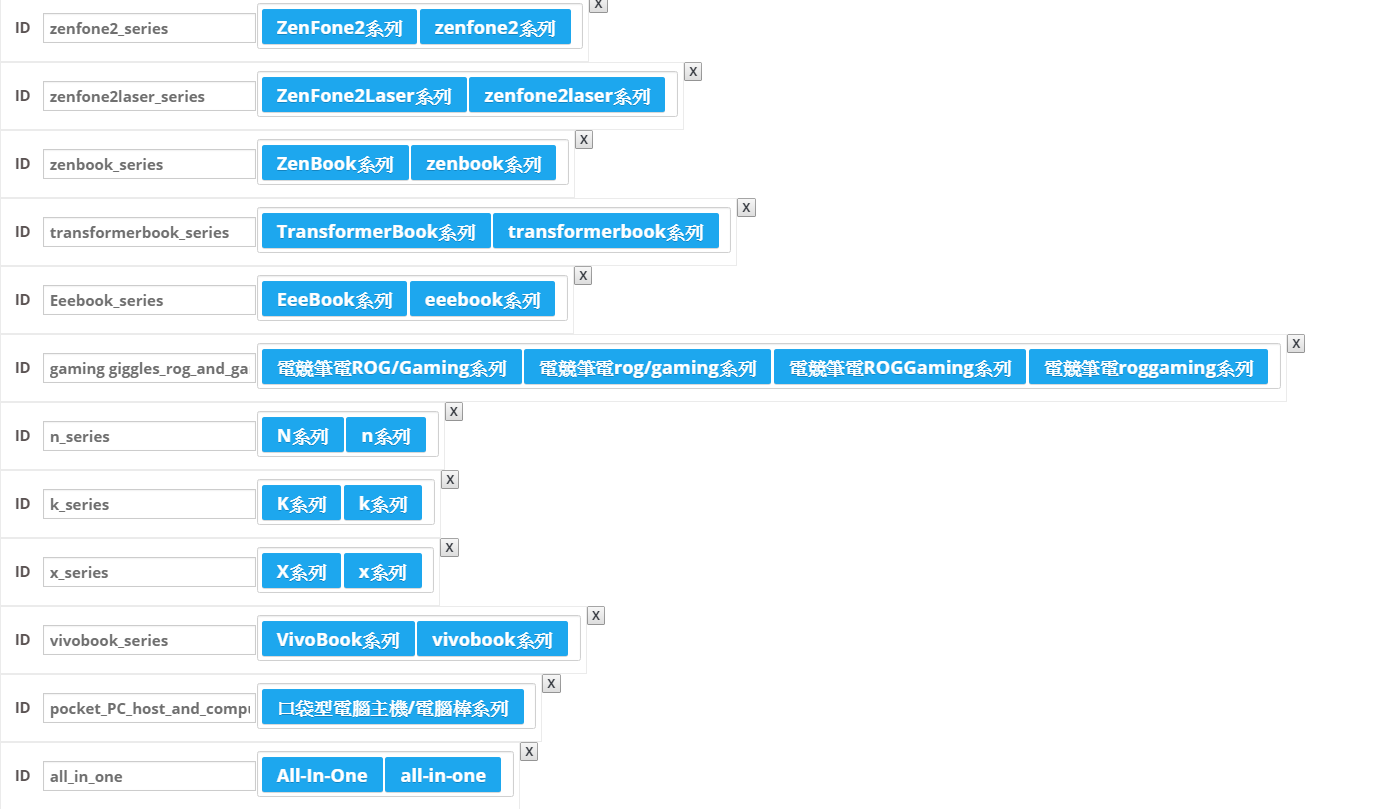
3 Class and 100 Instances in total.



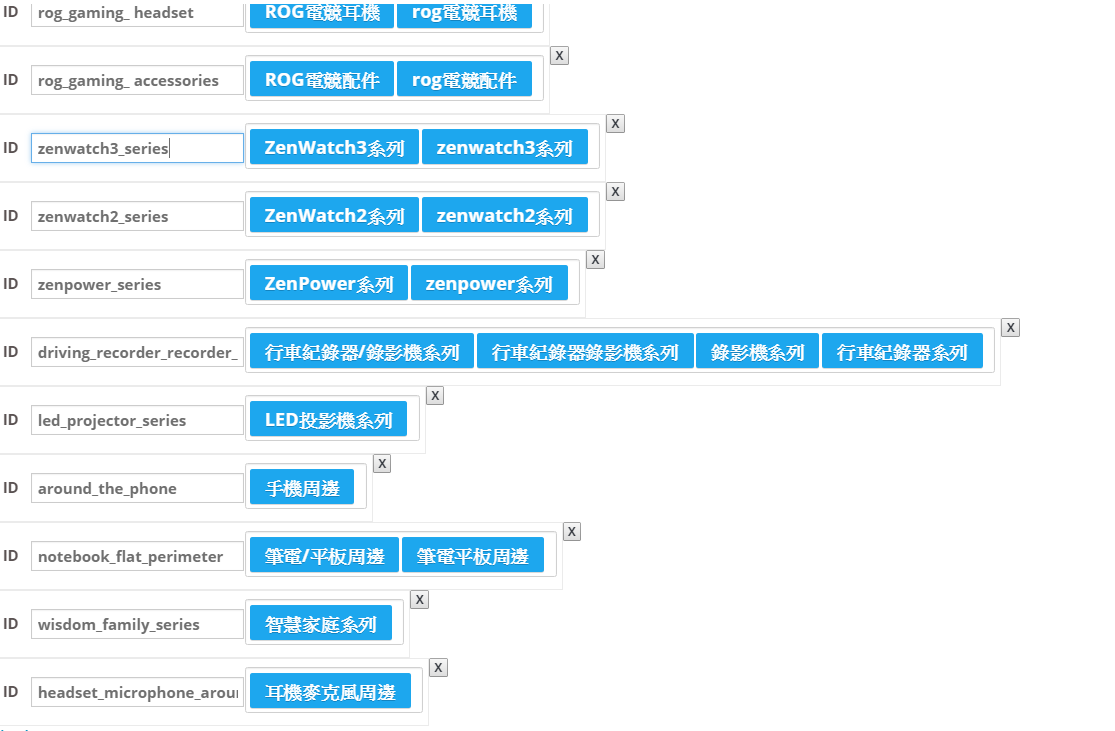












## Intent generation

請推薦TransformerBook系列商品有什麼

請推薦ZenWatch3系列有什麼商品

請告訴我EeeBook系列有哪些商品

請告訴我ZenFone3Zoom系列的商品有哪些

請問ZenBook系列的商品有哪些

請問N系列有哪些商品

請介紹K系列有哪些商品

請介紹X系列商品有什麼

VivoBook系列的商品有什麼

EeeBook系列有什麼商品

請推薦電競筆電All-In-One系列商品

請告訴我電競筆電K31桌上型電腦系列的商品

請問電競筆電ROG/Gaming系列的商品

I-ask.Product. 裡有 2項如下



請問ROG FX553VD的價格是多少



ASUS Transformer Book的價格是多少

Our training Corus(trainnew.txt) and testing corpus(testnew.txt)

Are put in the following system:

https://github.com/HsiaoyenChuang/chatbot/trainnew.txt

# KEY FUNCTIONS

* **Zenbo SDK for Intent /Slot Classification.** Detect language of utterance and pre-process n-best speech recognition result. And find the best result of Intents.

Dialogue system provides intents and corresponding parameters by sending interaction Json to Client. Following is the sample using editor as following site:

<https://stage-developer.asus.com/tools/ds-editor.jsp>

Following results from editor show the result sent from Dialogue system.

Question:請告訴我ZenFone2Laser系列有什麼商品

( ApplicationObject : commodity HowToKnow : what

Product : zenfone2laser\_series)

Answers:

|  |
| --- |
|  |

* **Jointed model for Intent /Slot Classification:**

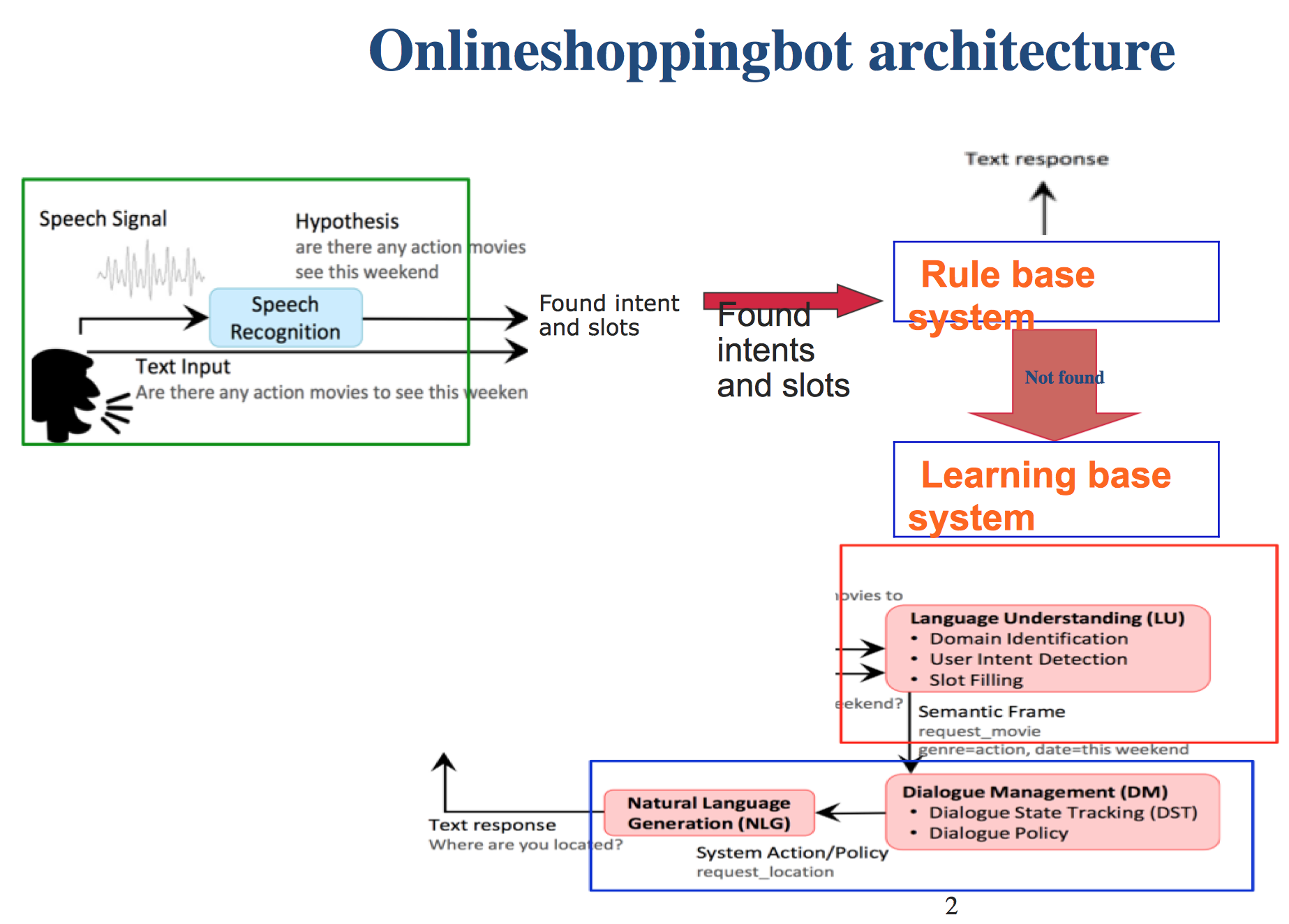
請問ASUS ZenFone Zoom 5.5吋價格多少

Result:

|  |
| --- |
| {'slot': ['o', 'o', 'S-Product.S-HowToKnow', 'S-Product.S-Price'], 'intent': 'I-Price'} |

Currently Jointed model has poor performance. We think it results from small corpus. We will improve this part later.

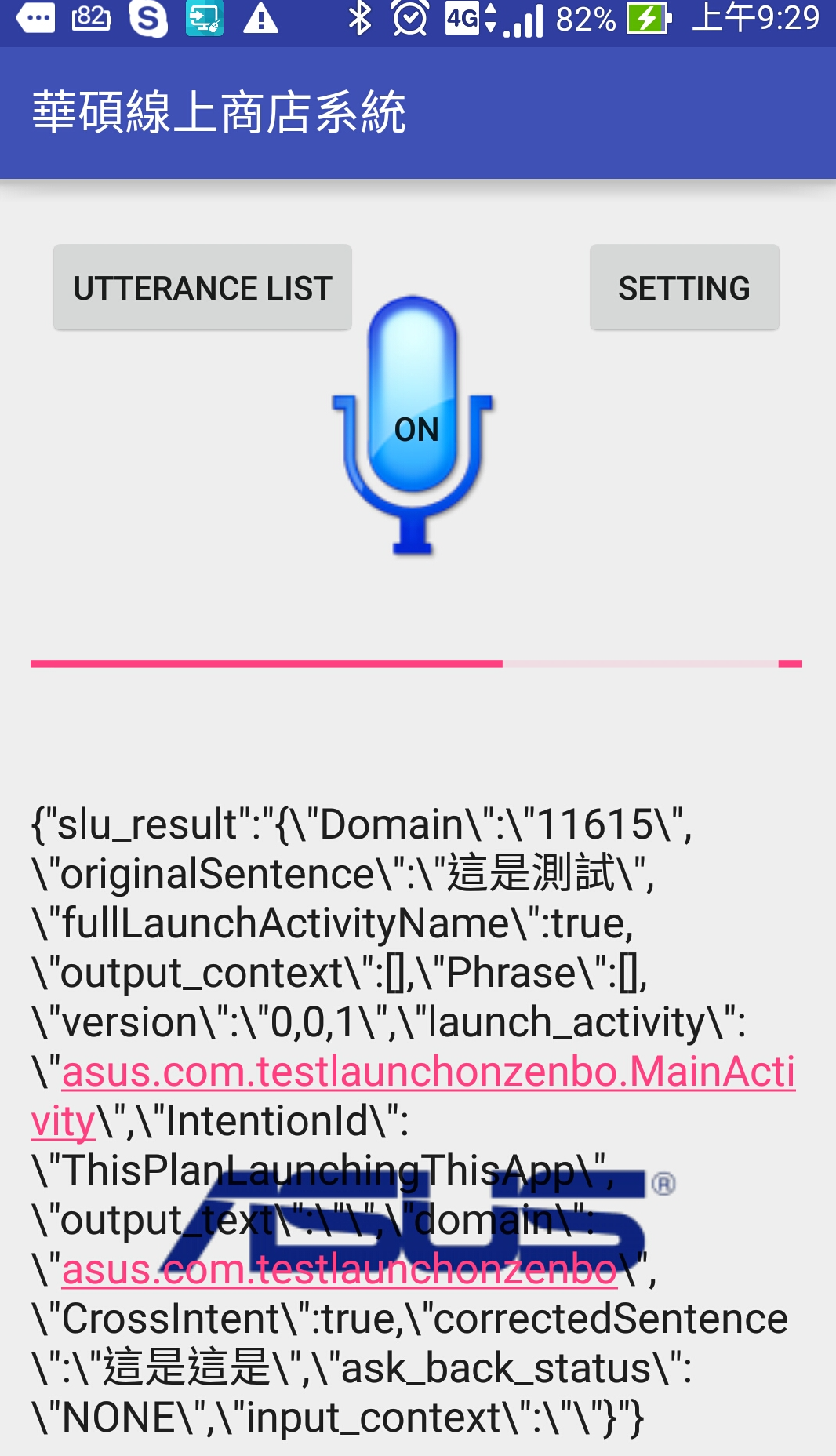
# SOFTWARE ARCHITECTURE

****

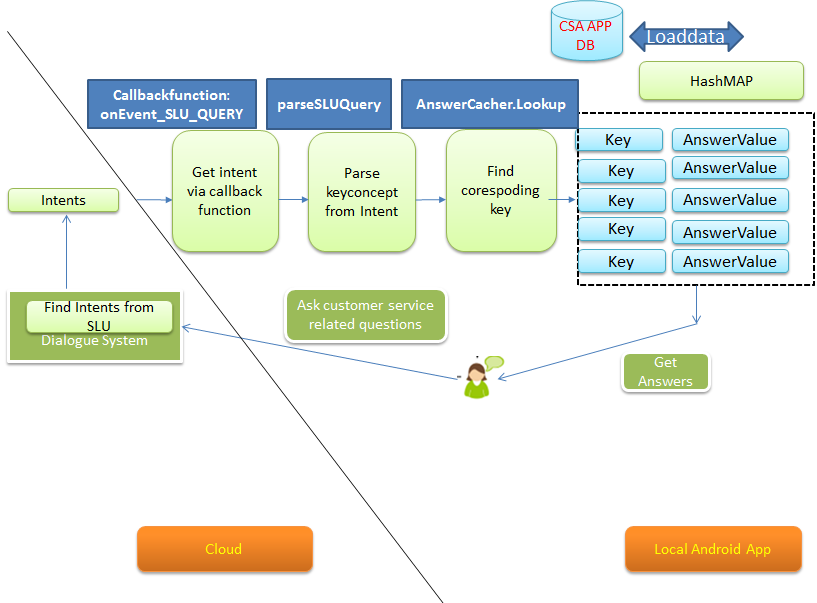
1. Android app to get voice signal or text from users
2. Check if rule-based system get the intent/slots or not
3. if yes, rule-based system handle it and response to user directly

if not, call web API and return json(for example ({'slot': ['o', 'o', 'S-Product.S-HowToKnow', 'S-Product.S-Price'], 'intent': 'I-Price'}) to illustrate the intents and slots of this sentence

* Android application to get text input and voice input

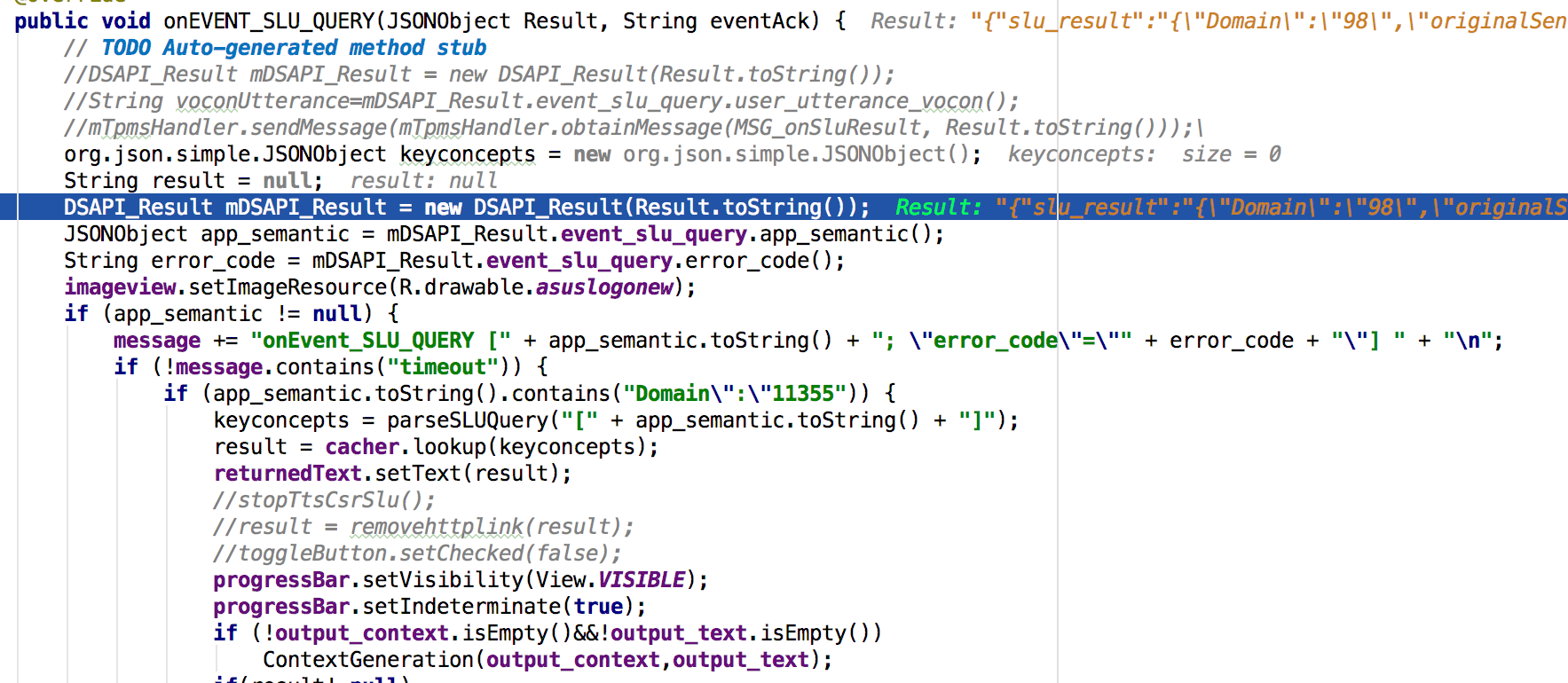


* Rule Based System
* Questions and responses are stored in online shop App DB and we extract its keys/values from AppDB and store in hash map for quick lookup and easy construction.
* Note: Current design uses hash map to simplify the implementation effort .I think current implementation can be placed into server side. DB can be replaced with sqlite or mysql relational DB per request.



Get data from Dialogue system in the **onEvent\_SLU\_QUERY** event callback function.

****



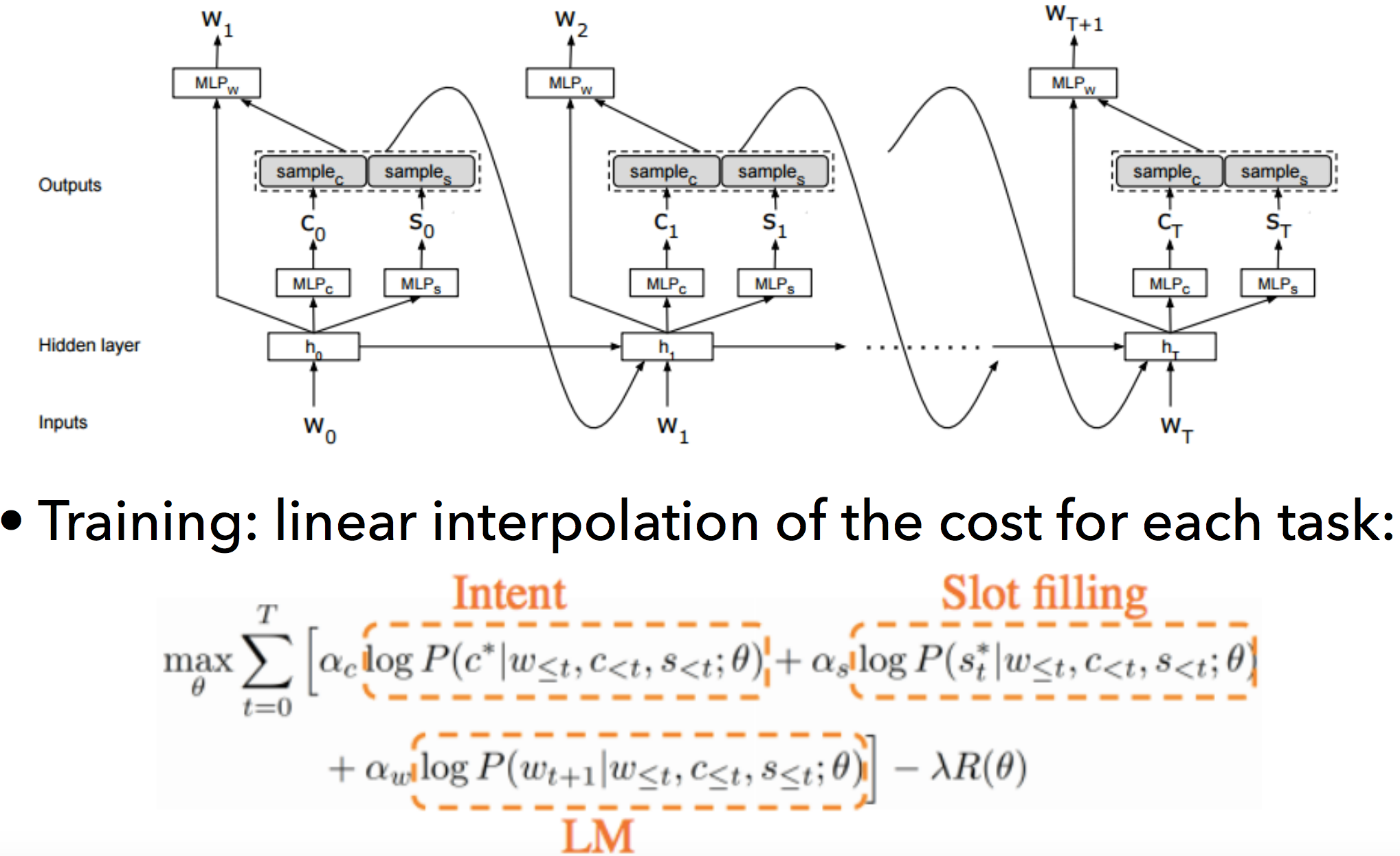
|  |  |
| --- | --- |
| Return Json format of Intent and Intents: | {Domain: Intentid,Slots } |

* Learning Based System

Use joint model from following paper.

Joint Online Spoken Language Understanding and Language Modeling

With Recurrent Neural Networks (https://arxiv.org/abs/1609.01462)



|  |  |
| --- | --- |
| API format | API format :/api/intent?text=xxxx   HTTP get approach |
| Return Json format of Intent and Intents: | {'slot': ['o', 'o', 'S-Product.S-HowToKnow', 'S-Product.S-Price'], 'intent': 'I-Price'} |

## Program List

List functionalities of each file

### AsusServiceMainActivity.java

* Initialize activity and get event using DSAPI.
* Use keyconcepts of intents to search answers.
* Implement ParseSLUQuery function

### AnswerCacher.java

* Extend AbstractGoldAnswerCacher class
* Implement lookup function
* Implement loaddata function

### AbstractGoldAnswerCacher.java

* Implement lookup function
* Implement hash map using abstract class

## API Interface

|  |  |
| --- | --- |
| **Return value** |  |
| **abstract** O | lookup(I input){  use key concept to lookup hash table and find answers |
| **JSONObject** result | ParseSLUQuery (String message)  Parse intents from DS and get key concopets |
| **Void** | loadData(String dataPath)  Load data from database and put them into hashmap. |
| **Void** | OnEVENT\_SLU\_QUERY(JSONObject Result, String eventAck)  Get event callback from dialogue system |