# Content

# 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**(Including many 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)

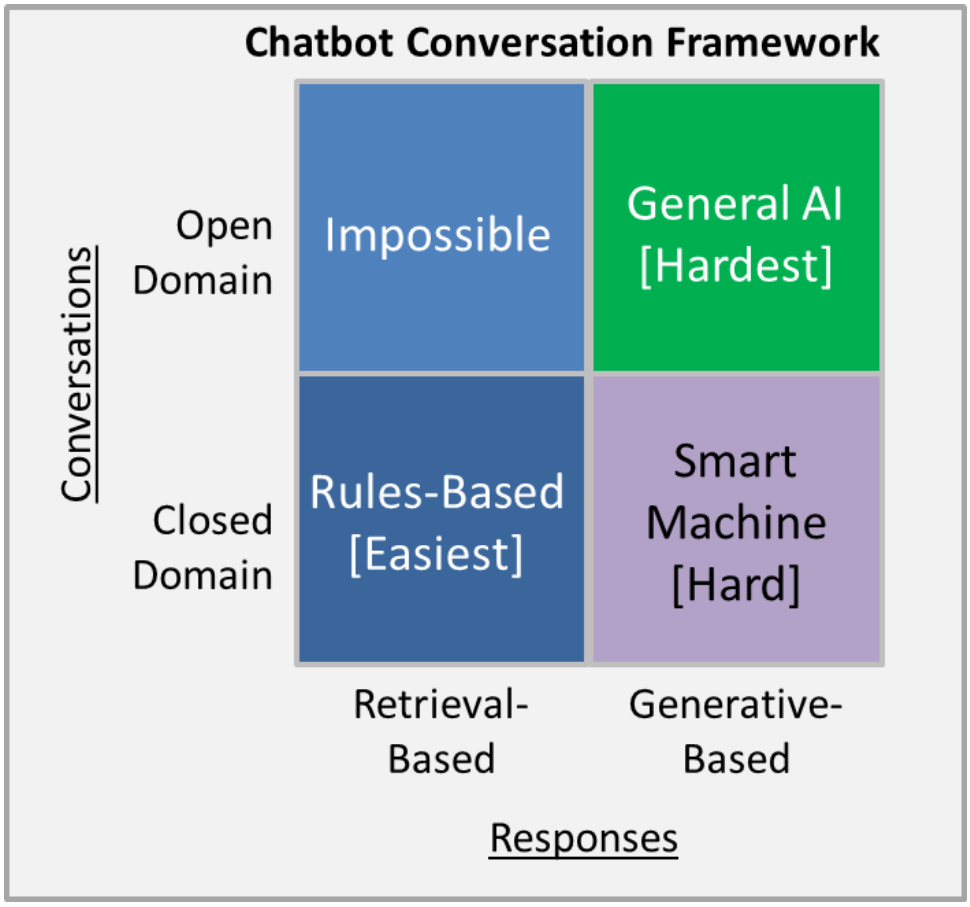
# Revision History

| 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 |

# 1. INTRODUCTION

## 1.1 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**

  - Supports multi-domain lexicons(Crowdsourcing ontology definition)
  - Support error correction of CSR results (based on existing domain ontologies to find the word that best fit Syllables
  - Find intentions and Appid for specific app

  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**

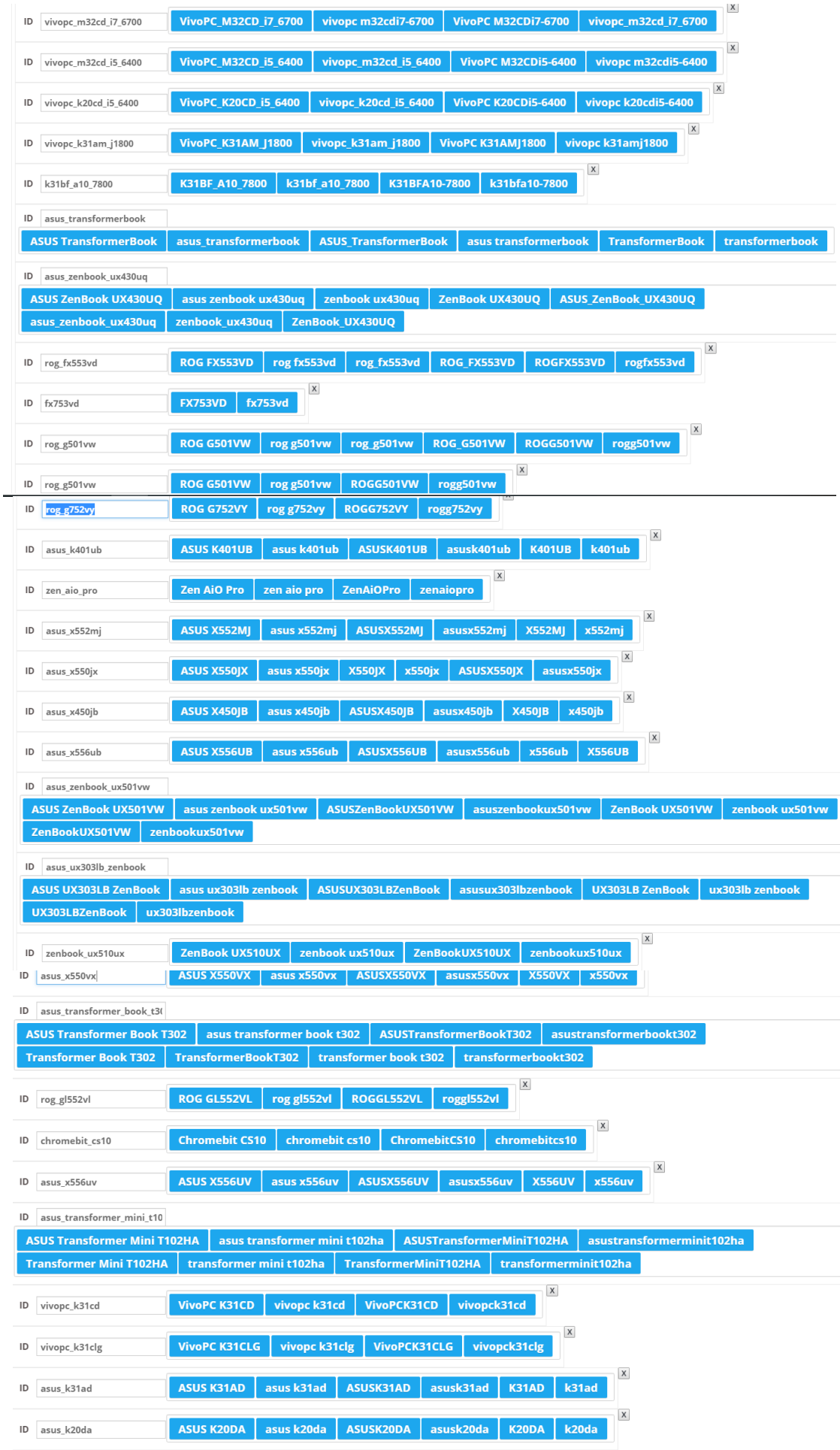
In the rum time, for an utterance which is recognized by speech recognition module, SLU outputs the best intentions and the related information from the knowledge database. SYSTEM ARCHITECTURE

## 1.2 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

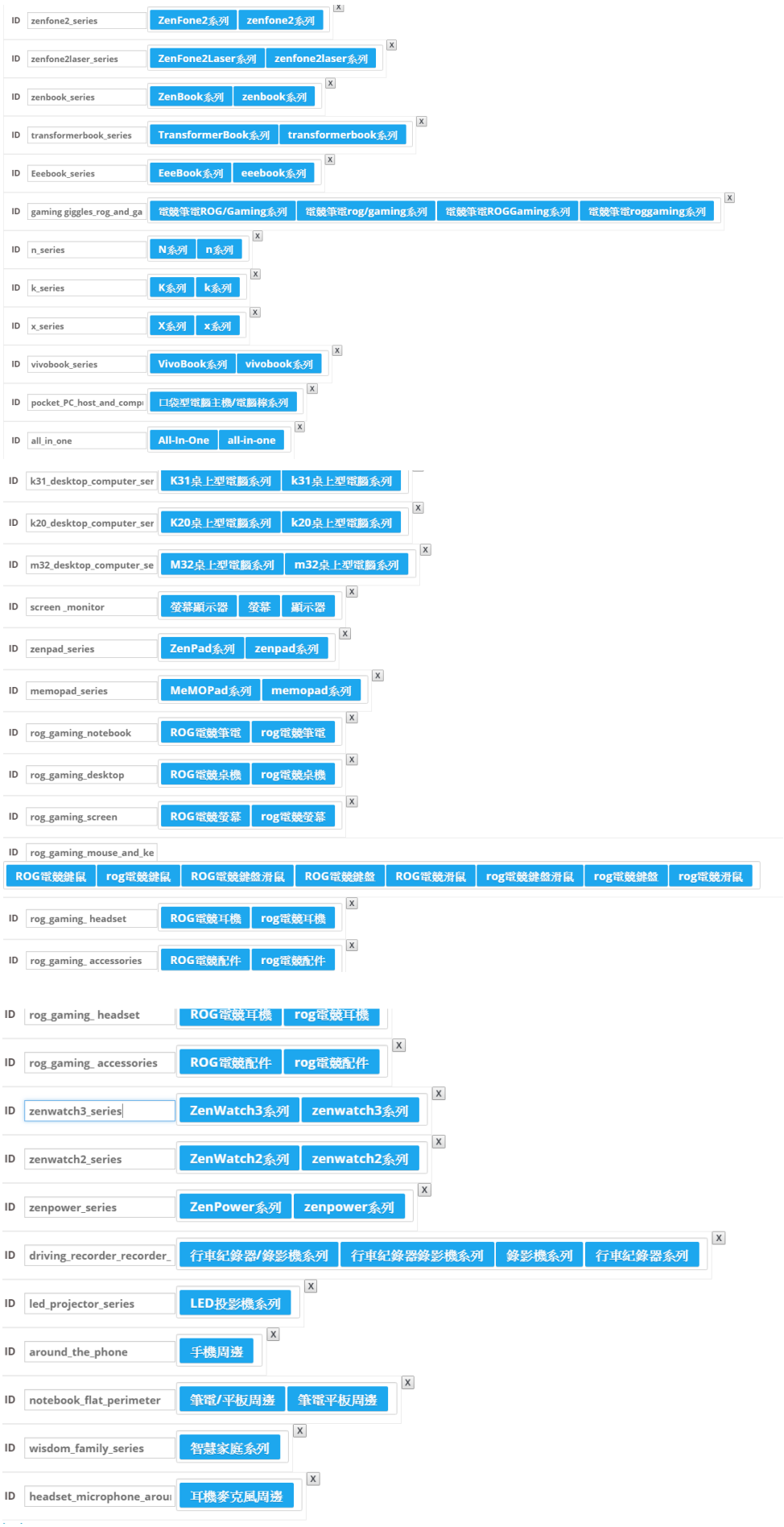
- **Product**
- **HowToKnow**
- **ApplicationObject**

3 Class and 100 Instances in total.


ID vivopc_m32cd_i7_6700
VivoPC_M32CD_i7_6700 vivopc m32cdi7-6700 VivoPC M32CDi7-6700 vivopc_m32cd_i7_6700
ID vivopc_m32cd_i5_6400
VivoPC_M32CD_i5_6400 vivopc_m32cd_i5_6400 VivoPC M32CDi5-6400 vivopc m32cdi5-6400
ID vivopc_k20cd_i5_6400
VivoPC_K20CD_i5_6400 vivopc_k20cd_i5_6400 VivoPC K20CDi5-6400 vivopc k20cdi5-6400
ID vivopc_k31am_j1800
VivoPC_K31AM_J1800 vivopc_k31am_j1800 VivoPC K31AMJ1800 vivopc k31amj1800
ID k31bf_a10_7800
K31BF_A10_7800 k31bf_a10_7800 K31BFA10-7800 k31bfa10-7800
ID asus_transformerbook
ASUS TransformerBook asus_transformerbook ASUS_TransformerBook asus transformerbook TransformerBook transformerbook
ID asus_zenbook_ux430uq
ASUS ZenBook UX430UQ asus zenbook ux430uq zenbook ux430uq ZenBook UX430UQ ASUS_ZenBook_UX430UQ
asus_zenbook_ux430uq zenbook_ux430uq ZenBook_UX430UQ
ID rog_fx553vd
ROG FX553VD rog fx553vd rog_fx553vd ROG_FX553VD ROGFX553VD rogfx553vd
ID fx753vd
FX753VD fx753vd
ID rog_g501vw
ROG G501VW rog g501vw rog_g501vw ROG_G501VW ROGG501VW rogg501vw
ID rog_g501vw
ROG G501VW rog g501vw ROGG501VW rogg501vw
ID rog_g752vy
ROG G752VY rog g752vy ROGG752VY rogg752vy
ID asus_k401ub
ASUS K401UB asus k401ub ASUSK401UB asusk401ub K401UB k401ub
ID zen_aio_pro
Zen AiO Pro zen aio pro ZenAiOPro zenaiopro
ID asus_x552mj
ASUS X552MJ asus x552mj ASUSX552MJ asusx552mj X552MJ x552mj
ID asus_x550jx
ASUS X550JX asus x550jx X550JX x550jx ASUSX550JX asusx550jx
ID asus_x450jb
ASUS X450JB asus x450jb ASUSX450JB asusx450jb X450JB x450jb
ID asus_x556ub
ASUS X556UB asus x556ub ASUSX556UB asusx556ub x556ub X556UB
ID asus_zenbook_ux501vw
ASUS ZenBook UX501VW asus zenbook ux501vw ASUSZenBookUX501VW asuszenbookux501vw ZenBook UX501VW zenbook ux501vw
ZenBookUX501VW zenbookux501vw
ID asus_ux303lb_zenbook
ASUS UX303LB ZenBook asus ux303lb zenbook ASUSUX303LBZenBook asusux303lbzenbook UX303LB ZenBook ux303lb zenbook
UX303LBZenBook ux303lbzenbook
ID zenbook_ux510ux
ZenBook UX510UX zenbook ux510ux ZenBookUX510UX zenbookux510ux
ID asus_x550vx
ASUS X550VX asus x550vx ASUSX550VX asusx550vx X550VX x550vx
ID asus_transformer_book_t3(
ASUS Transformer Book T302 asus transformer book t302 ASUSTransformerBookT302 asustransformerbookt302
Transformer Book T302 TransformerBookT302 transformer book t302 transformerbookt302
ID rog_gl552vl
ROG GL552VL rog gl552vl ROGGL552VL roggl552vl
ID chromebit_cs10
Chromebit CS10 chromebit cs10 ChromebitCS10 chromebitcs10
ID asus_x556uv
ASUS X556UV asus x556uv ASUSX556UV asusx556uv X556UV x556uv
ID asus_transformer_mini_t10
ASUS Transformer Mini T102HA asus transformer mini t102ha ASUSTransformerMiniT102HA asustransformerminit102ha
Transformer Mini T102HA transformer mini t102ha TransformerMiniT102HA transformerminit102ha
ID vivopc_k31cd
VivoPC K31CD vivopc k31cd VivoPCK31CD vivopck31cd
ID vivopc_k31clg
VivoPC K31CLG vivopc k31clg VivoPCK31CLG vivopck31clg
ID asus_k31ad
ASUS K31AD asus k31ad ASUSK31AD asusk31ad K31AD k31ad
ID asus_k20da
ASUS K20DA asus k20da ASUSK20DA asusk20da K20DA k20da


ID asus_vivopc_k31ad
ASUS VivoPC K31AD
asus vivopc k31ad
ASUSVivoPCK31AD
asusvivopck31ad
VivoPC K31AD
VivoPCK31AD
vivopc k31ad
vivopck31ad
ID vivo_pc_k31da
Vivo PC K31DA
vivo pc k31da
VivoPCK31DA
vivopck31da
ID asus_m32cd
ASUS M32CD
asus m32cd
ASUSM32CD
asusm32cd
M32CD
m32cd
ID asus_zenfone_3_5.5incM32(
ASUS ZenFone 3 5.5吋
asus zenfone 3 5.5吋
ASUSZenFone35.5吋
asuszenfone35.5吋
ZenFone35.5吋
zenfone35.5吋
ID zenfone_3_deluxe
ZenFone 3 Deluxe
zenfone 3 deluxe
ID asus_zenfone_3_5.2inch
ASUS ZenFone 3 5.2吋
asus zenfone 3 5.2吋
ID zenfone_3_zoom_5.5inch
ZenFone 3 Zoom 5.5吋
zenfone 3 zoom 5.5吋
ID asus_zenfone_3_ultra_6.8in
ASUS ZenFone 3 Ultra 6.8吋
asus zenfone 3 ultra 6.8吋
ID zenfone_3_laser
ZenFone 3 Laser
zenfone 3 laser
ID zenfone_3_max
ZenFone 3 Max
zenfone 3 max
ID zenfone_live_5inch
ZenFone Live 5吋
zenfone live 5吋


ID asus_padfone_mini
ASUS PadFone mini
asus padfone mini
ID asus_zenfone_2_ze551ml_5.
ASUS ZenFone 2 ZE551ML 5.5吋
asus zenfone 2 ze551ml 5.5吋
ID asus_zenfone_zoom_5.5incl
ASUS ZenFone Zoom 5.5吋
asus zenfone zoom 5.5吋
ID asus_zenfone_2_laser_6inch
ASUS ZenFone 2 Laser 6吋
asus zenfone 2 laser 6吋
ID asus_zenfone_go_tv
ASUS ZenFone Go TV
asus zenfone go tv
ID zenfone_2_deluxe_5.5inch
ZenFone 2 Deluxe 5.5吋
zenfone 2 deluxe 5.5吋
ID new_asus_zenpad_10
NEW ASUS ZenPad 10
new asus zenpad 10
ID asus_zenpad_c_7.0_wifi
ASUS ZenPad C 7.0WiFi版
asus zenpad c 7.0wifi版
ID asus_zenpad_s_8.0_wifi
ASUS ZenPad S 8.0 WiFi版
asus zenpad s 8.0 wifi版
ID new_asus_zenpad_8.0
NEW ASUS ZenPad 8.0
new asus zenpad 8.0
ID asus_zenwatch_2
ASUS ZenWatch 2
asus zenwatch 2
ID zenbo_charging_seat
Zenbo充電座
zenbo充電座
ID zenb_standard_edition
Zenbo標準版
zenbo標準版
ID zenbo_luxury_value_versio
Zenbo豪華超值版
zenbo豪華超值版
ID zenfone3_series
ZenFone3美型機系列
zenfone3美型機系列
ZenFone3系列
zenfone3系列
ID zenfone3zoom_series
ZenFone3Zoom系列
zenfone3zoom系列
ID zenfone3max_series
ZenFone3Max系列
zenfone3max系列
ID zenfone3deluxe_series
ZenFone3Deluxe系列
zenfone3deluxe系列
ID zenfone3ultra_series
ZenFone3Ultra系列
zenfone3ultra系列
ID zenfone3laser_series
ZenFone3Laser系列
zenfone3laser系列
ID zenfonelive_series
ZenFoneLive系列
zenfonelive系列


ID zenfone2_series
ZenFone2系列
zenfone2系列
ID zenfone2laser_series
ZenFone2Laser系列
zenfone2laser系列
ID zenbook_series
ZenBook系列
zenbook系列
ID transformerbook_series
TransformerBook系列
transformerbook系列
ID Eeebook_series
EeeBook系列
eeebook系列
ID gaming giggles_rog_and_ga
電競筆電ROG/Gaming系列
電競筆電rog/gaming系列
電競筆電ROGGaming系列
電競筆電roggaming系列
ID n_series
N系列
n系列
ID k_series
K系列
k系列
ID x_series
X系列
x系列
ID vivobook_series
VivoBook系列
vivobook系列
ID pocket_PC_host_and_compu
口袋型電腦主機/電腦棒系列
ID all_in_one
All-In-One
all-in-one
ID k31_desktop_computer_ser
K31桌上型電腦系列
k31桌上型電腦系列
ID k20_desktop_computer_ser
K20桌上型電腦系列
k20桌上型電腦系列
ID m32_desktop_computer_se
M32桌上型電腦系列
m32桌上型電腦系列
ID screen _monitor
螢幕顯示器
螢幕
顯示器
ID zenpad_series
ZenPad系列
zenpad系列
ID memopad_series
MeMOPad系列
memopad系列
ID rog_gaming_notebook
ROG電競筆電
rog電競筆電
ID rog_gaming_desktop
ROG電競桌機
rog電競桌機
ID rog_gaming_screen
ROG電競螢幕
rog電競螢幕
ID rog_gaming_mouse_and_ke
ROG電競鍵鼠
rog電競鍵鼠
ROG電競鍵盤滑鼠
ROG電競鍵盤
ROG電競滑鼠
rog電競鍵盤滑鼠
rog電競鍵盤
rog電競滑鼠
ID rog_gaming_ headset
ROG電競耳機
rog電競耳機
ID rog_gaming_ accessories
ROG電競配件
rog電競配件
ID rog_gaming_ headset
ROG電競耳機
rog電競耳機
ID rog_gaming_ accessories
ROG電競配件
rog電競配件
ID zenwatch3_series
ZenWatch3系列
zenwatch3系列
ID zenwatch2_series
ZenWatch2系列
zenwatch2系列
ID zenpower_series
ZenPower系列
zenpower系列
ID driving_recorder_recorder_
行車紀錄器/錄影機系列
行車紀錄器錄影機系列
錄影機系列
行車紀錄器系列
ID led_projector_series
LED投影機系列
ID around_the_phone
手機周邊
ID notebook_flat_perimeter
筆電/平板周邊
筆電平板周邊
ID wisdom_family_series
智慧家庭系列
ID headset_microphone_arou
耳機麥克風周邊

## 1.3 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

## 2. 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

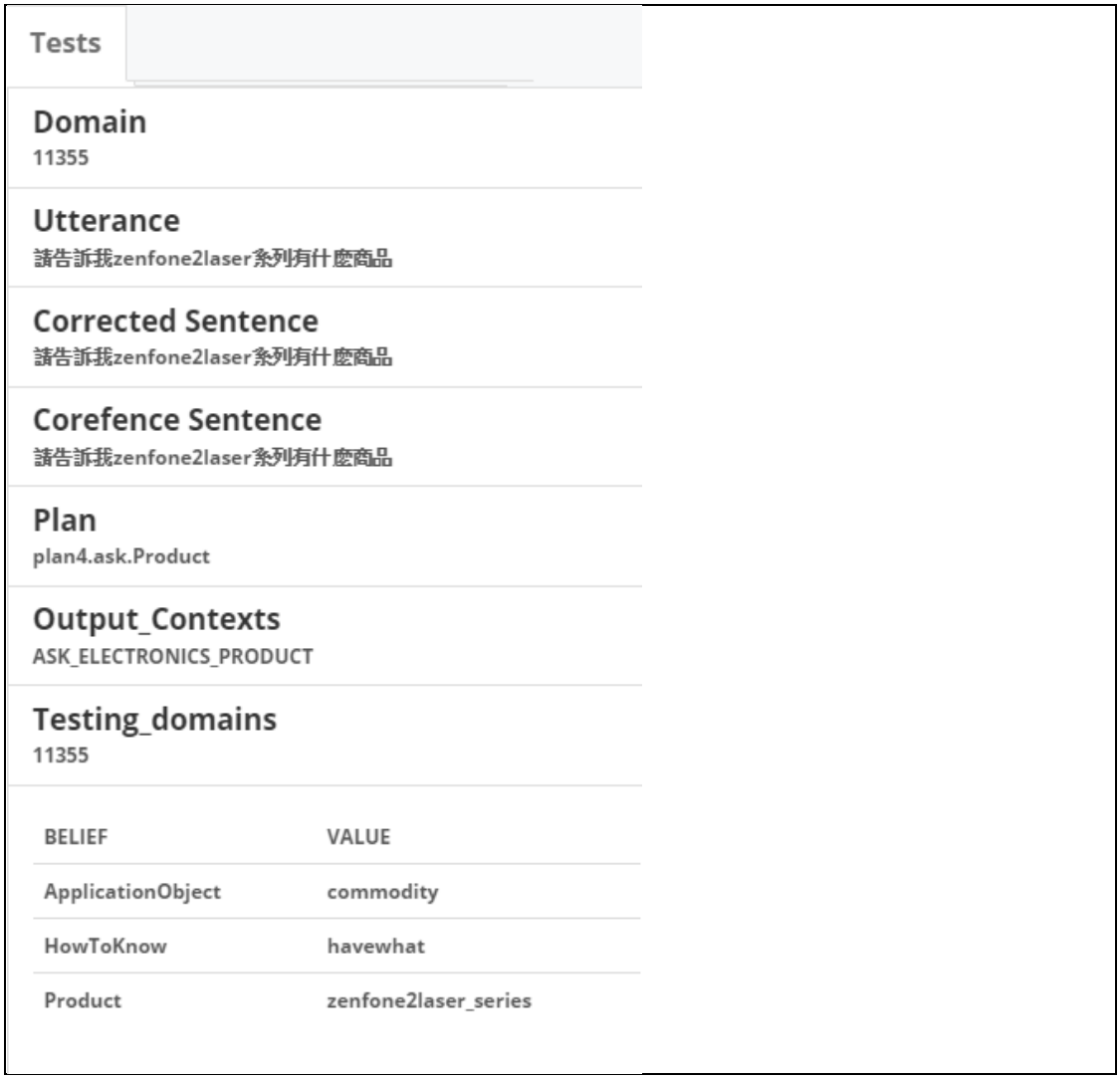
Following results from editor show the result sent from Dialogue system.

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

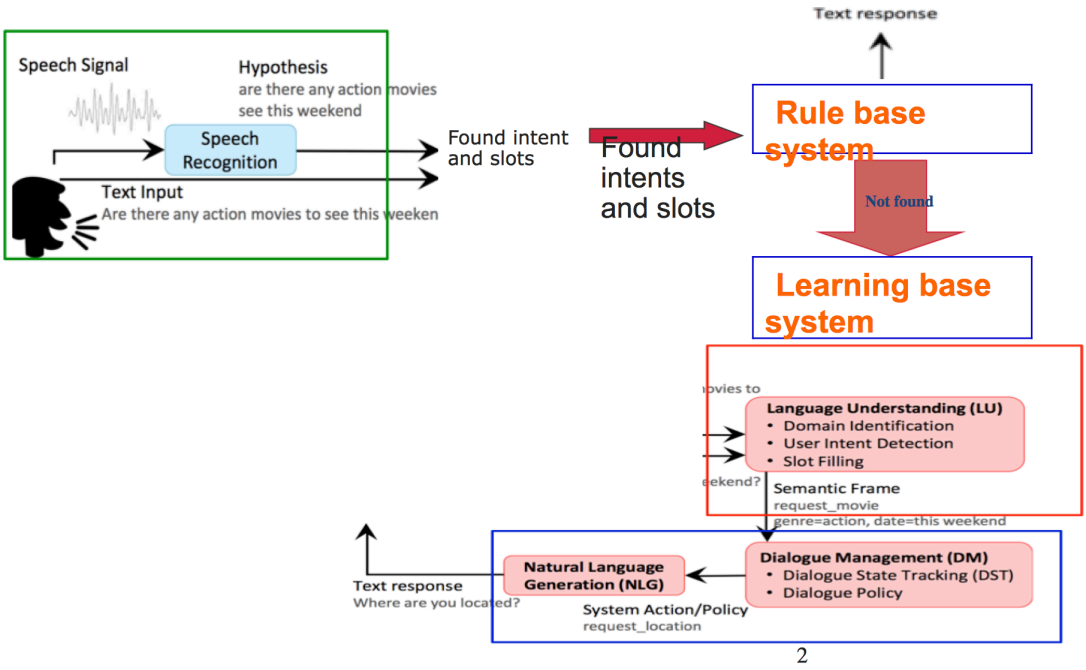
  Result:

  ```
  {'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.

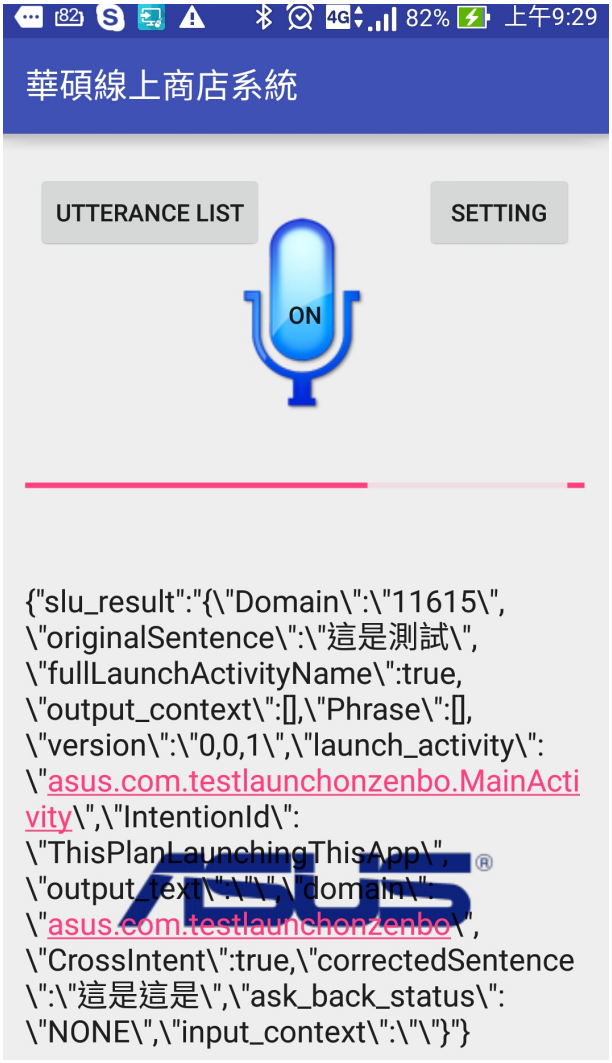
# 3. SOFTWARE ARCHITECTURE

## Onlineshoppingbot architecture



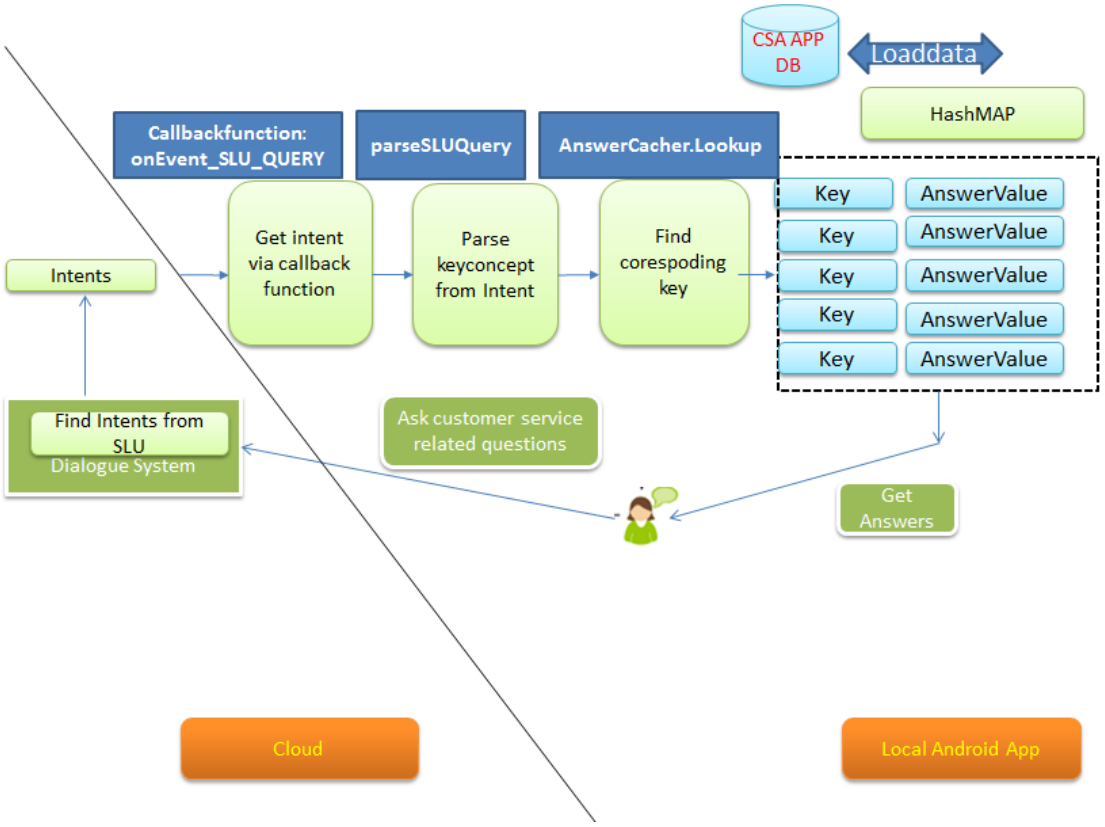

0) Android app to get voice signal or text from users
1) Check if rule-based system get the intent/slots or not
2) if yes, rule-based system handle it and response to user directly
   if not, call web API and return json(for example ({'slot': ['o', 'o', 'S-Product.S-HowToKnow', 'S-Product.S-Price'], '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.

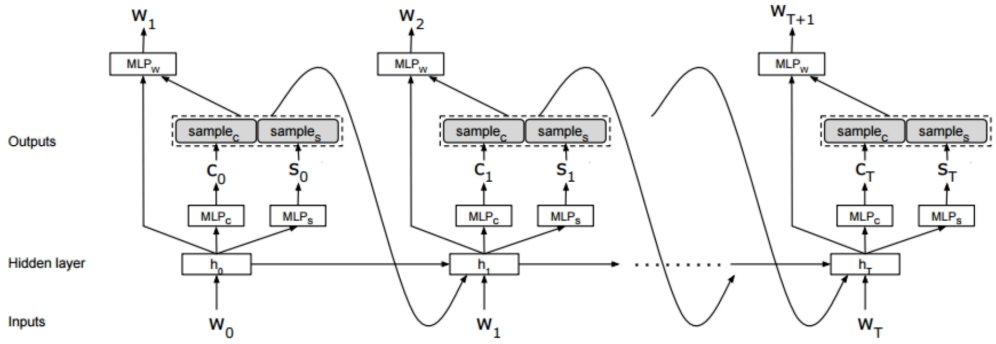
```
/*
Comments : onEVENT_SLU_QUERY Is the event callback function of DSAPI .
Result of dialogue system is stored in app_semantic.
ParseSLUQuery :parses intents and get key concepts of intents
*/
@Override
public void onEVENT_SLU_QUERY(JSONObject Result, String eventAck) {
    // TODO Auto-generated method stub
    //DSAPI_Result mDSAPI_Result = new DSAPI_Result(Result.toString());
    //String voconUtterance=mDSAPI_Result.event_slu_query.user_utterance_vocon();
    //mTpmsHandler.sendMessage(mTpmsHandler.obtainMessage(MSG_onSluResult, Result.toString()));\
    org.json.simple.JSONObject keyconcepts = new  org.json.simple.JSONObject();
    DSAPI_Result mDSAPI_Result = new DSAPI_Result(Result.toString());
    JSONObject app_semantic = mDSAPI_Result.event_slu_query.app_semantic();
    String error_code = mDSAPI_Result.event_slu_query.error_code();
    if (app_semantic != null) {
        // message += "onEvent_SLU_QUERY [" + app_semantic.toString() + "; \"error_code\"=\"" + error_code + "\"] " + "\n";
        if ( app_semantic.toString().contains("Domain\":\"11337")) {
            keyconcepts = parseSLUQuery("[" + app_semantic.toString() + "]");
            String result = cacher.lookup(keyconcepts);
            returnedText.setText(result);
        }
    }
    else {
        // message += "onEvent_SLU_QUERY [" + "\"error_code\"=\"" + error_code + "\"] " + "\n";
    }

    //last event ??? stop progress bar
    toggleButton.setChecked(false);
    progressBar.setVisibility(View.INVISIBLE);
    progressBar.setIndeterminate(false);
}
```

```
public void onEVENT_SLU_QUERY(JSONObject Result, String eventAck) {  Result: "{"slu_result":"{\"Domain\":\"98\",\"originalSen
    // TODO Auto-generated method stub
    //DSAPI_Result mDSAPI_Result = new DSAPI_Result(Result.toString());
    //String voconUtterance=mDSAPI_Result.event_slu_query.user_utterance_vocon();
    //mTpmsHandler.sendMessage(mTpmsHandler.obtainMessage(MSG_onSluResult, Result.toString()));\
    org.json.simple.JSONObject keyconcepts = new org.json.simple.JSONObject();  keyconcepts:  size = 0
    String result = null;  result: null
    DSAPI_Result mDSAPI_Result = new DSAPI_Result(Result.toString());  Result: "{"slu_result":"{\"Domain\":\"98\",\"originalS
    JSONObject app_semantic = mDSAPI_Result.event_slu_query.app_semantic();
    String error_code = mDSAPI_Result.event_slu_query.error_code();
    imageview.setImageResource(R.drawable.asuslogonew);
    if (app_semantic != null) {
        message += "onEvent_SLU_QUERY [" + app_semantic.toString() + "; \"error_code\"=\"" + error_code + "\"] " + "\n";
        if (!message.contains("timeout")) {
            if (app_semantic.toString().contains("Domain\":\"11355")) {
                keyconcepts = parseSLUQuery("[" + app_semantic.toString() + "]");
                result = cacher.lookup(keyconcepts);
                returnedText.setText(result);
                //stopTtsCsrSlu();
                //result = removehttplink(result);
                //toggleButton.setChecked(false);
                progressBar.setVisibility(View.VISIBLE);
                progressBar.setIndeterminate(true);
                if (!output_context.isEmpty()&&!output_text.isEmpty())
                    ContextGeneration(output_context,output_text);
```

| | |
|---|---|
| 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)




• Training: linear interpolation of the cost for each task:

$$\max_{\theta} \sum_{t=0}^{T} \Big[ \alpha_c \log P(c^*|w_{\leq t}, c_{<t}, s_{<t}; \theta) + \alpha_s \log P(s_t^*|w_{\leq t}, c_{<t}, s_{<t}; \theta) + \alpha_w \log P(w_{t+1}|w_{\leq t}, c_{\leq t}, s_{\leq t}; \theta) \Big] - \lambda R(\theta)$$

Intent — Slot filling — LM

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

### 3.1 Program List

List functionalities of each file

#### 3.1.1 AsusServiceMainActivity.java

- Initialize activity and get event using DSAPI.
- Use keyconcepts of intents to search answers.

- Implement ParseSLUQuery function

### 3.1.2 AnswerCacher.java

- Extend AbstractGoldAnswerCacher class
- Implement lookup function
- Implement loaddata function

### 3.1.3 AbstractGoldAnswerCacher.java

- Implement lookup function
- Implement hash map using abstract class

## 3.2 API Interface

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