

# Content

1. INTRODUCTION .....	3
1.1 Project Goal .....	3
1.2 Background knowledge.....	3
1.3 Concepts generation (Classes & Instances) .....	4
1.4 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**(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)

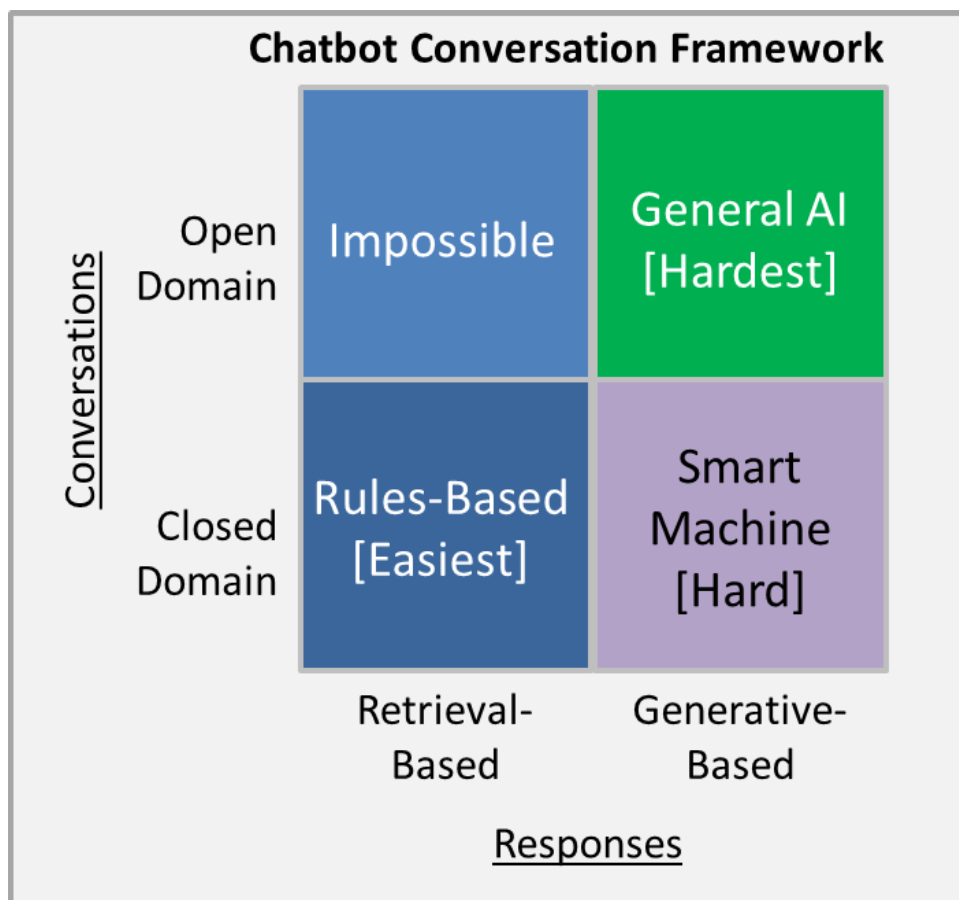
## Revision History

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 run 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

### **1.3 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	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_5.5incM32C	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	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 電競鍵盤	rog 電競鍵盤	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
----	------------------------	----------	----------

D	zenb_standard_edition	Zenbo標準版	zenbo標準版	X		
D	zenbo_luxury_value_version	Zenbo豪華超值版	zenbo豪華超值版	X		
D	zenfone3_series	ZenFone3美型機系列	zenfone3美型機系列	ZenFone3系列	zenfone3系列	X
D	zenfone3zoom_series	ZenFone3Zoom系列	zenfone3zoom系列	X		
D	zenfone3max_series	ZenFone3Max系列	zenfone3max系列	X		
D	zenfone3deluxe_series	ZenFone3Deluxe系列	zenfone3deluxe系列	X		
D	zenfone3ultra_series	ZenFone3Ultra系列	zenfone3ultra系列	X		
D	zenfone3laser_series	ZenFone3Laser系列	zenfone3laser系列	X		
D	zenfonelive_series	ZenFoneLive系列	zenfonelive系列	X		
D	zenfonego_series	ZenFoneGo系列	zenfonego系列	X		
D	zenfonezoom_series	ZenFoneZoom系列	zenfonezoom系列	X		
D	rog_gaming_headset	ROG電競耳機	rog電競耳機			
D	rog_gaming_accessories	ROG電競配件	rog電競配件	X		
D	zenwatch3_series	ZenWatch3系列	zenwatch3系列	X		
D	zenwatch2_series	ZenWatch2系列	zenwatch2系列	X		
D	zenpower_series	ZenPower系列	zenpower系列	X		
D	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	X
D	led_projector_series	LED投影機系列	X			
D	around_the_phone	手機周邊	X			
D	notebook_flat_perimeter	筆電/平板周邊	筆電平板周邊	X		
D	wisdom_family_series	智慧家庭系列	X			
D	headset_microphone_arou	耳機麥克風周邊	X			

## 1.4 Intent generation

請推薦 TransformerBook 系列商品有什麼

請推薦	Product : Product	HowToKnow : HowToKnow	ApplicationObject : ApplicationObject
-----	-------------------	-----------------------	---------------------------------------

請告訴我	Product : Product	HowToKnow : HowToKnow	ApplicationObject : ApplicationObject
------	-------------------	-----------------------	---------------------------------------

請告訴我	Product : Product	ApplicationObject : ApplicationObject	HowToKnow : HowToKnow
------	-------------------	---------------------------------------	-----------------------

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	Product : Product	ApplicationObject : ApplicationObject	HowToKnow : HowToKnow
----	-------------------	---------------------------------------	-----------------------

請問 ZenBook 系列的商品有哪些

請問	Product : Product	HowToKnow : HowToKnow	ApplicationObject : ApplicationObject
----	-------------------	-----------------------	---------------------------------------

請問 N 系列有哪些商品

請介紹	Product : Product	HowToKnow : HowToKnow	ApplicationObject : ApplicationObject
-----	-------------------	-----------------------	---------------------------------------

請介紹 K 系列有哪些商品

請介紹	Product : Product	ApplicationObject : ApplicationObject	HowToKnow : HowToKnow
-----	-------------------	---------------------------------------	-----------------------

請介紹 X 系列商品有什麼

Product : Product	ApplicationObject : ApplicationObject	HowToKnow : HowToKnow
-------------------	---------------------------------------	-----------------------

VivoBook 系列的商品有什麼

Product : Product	HowToKnow : HowToKnow	ApplicationObject : ApplicationObject
-------------------	-----------------------	---------------------------------------

EeeBook 系列有什麼商品

請推薦	Product : Product	ApplicationObject : ApplicationObject
-----	-------------------	---------------------------------------

請推薦電競筆電 All-In-One 系列商品

請告訴我	Product : Product	ApplicationObject : ApplicationObject
------	-------------------	---------------------------------------

請告訴我電競筆電 K31 桌上型電腦系列的商品

請問	Product : Product	ApplicationObject : ApplicationObject
----	-------------------	---------------------------------------

請問電競筆電 ROG/Gaming 系列的商品

I-ask.Product. 裡有 2 項如下

請問	Product : Product	HowToKnow : HowToKnow
----	-------------------	-----------------------

請問 ROG FX553VD 的價格是多少

**Product : Product**

**HowToKnow : HowToKnow**

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:



<b>Tests</b>	
<b>Domain</b> 11355	
<b>Utterance</b> 請告訴我zenfone2laser系列有什麼商品	
<b>Corrected Sentence</b> 請告訴我zenfone2laser系列有什麼商品	
<b>Coreference Sentence</b> 請告訴我zenfone2laser系列有什麼商品	
<b>Plan</b> plan4.ask.Product	
<b>Output_Contexts</b> ASK_ELECTRONICS_PRODUCT	
<b>Testing_domains</b> 11355	
<b>BELIEF</b>	<b>VALUE</b>
ApplicationObject	commodity
HowToKnow	havewhat
Product	zenfone2laser_series

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

請問 ASUS ZenFone Zoom 5.5 吋價格多少

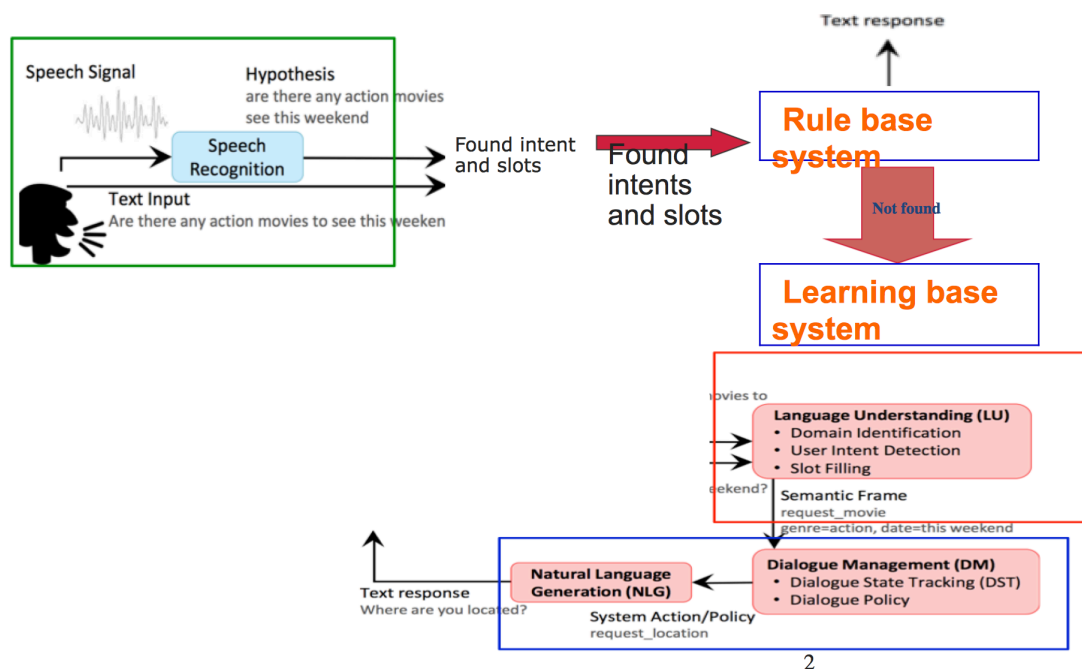
Result:

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

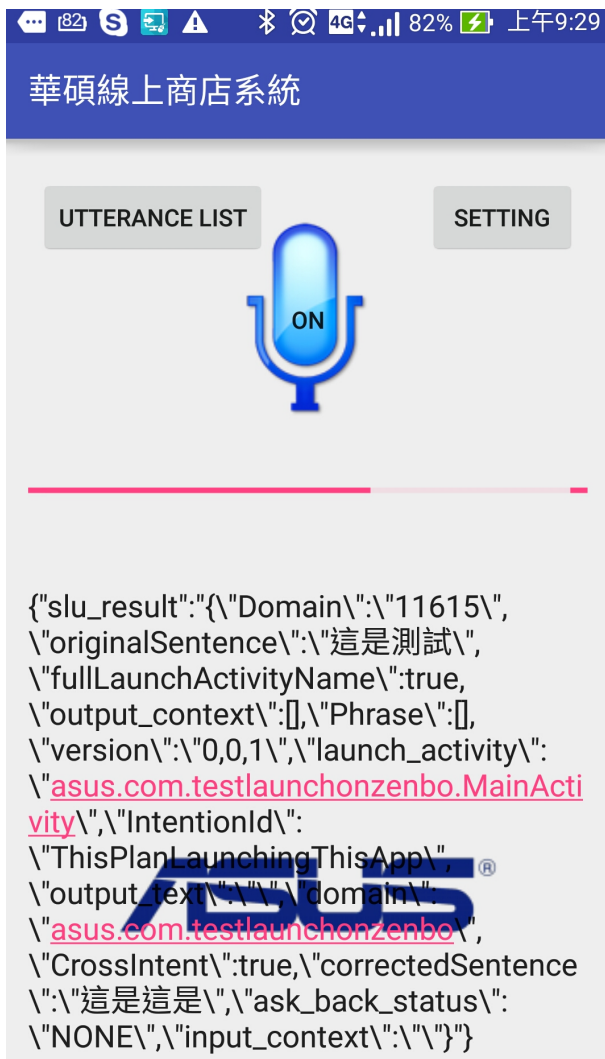
Currently Jointed model has poor performance. We think it results from small corpus. We will improve this part later.

### 3 SOFTWARE ARCHITECTURE

#### Onlineshoppingbot architecture

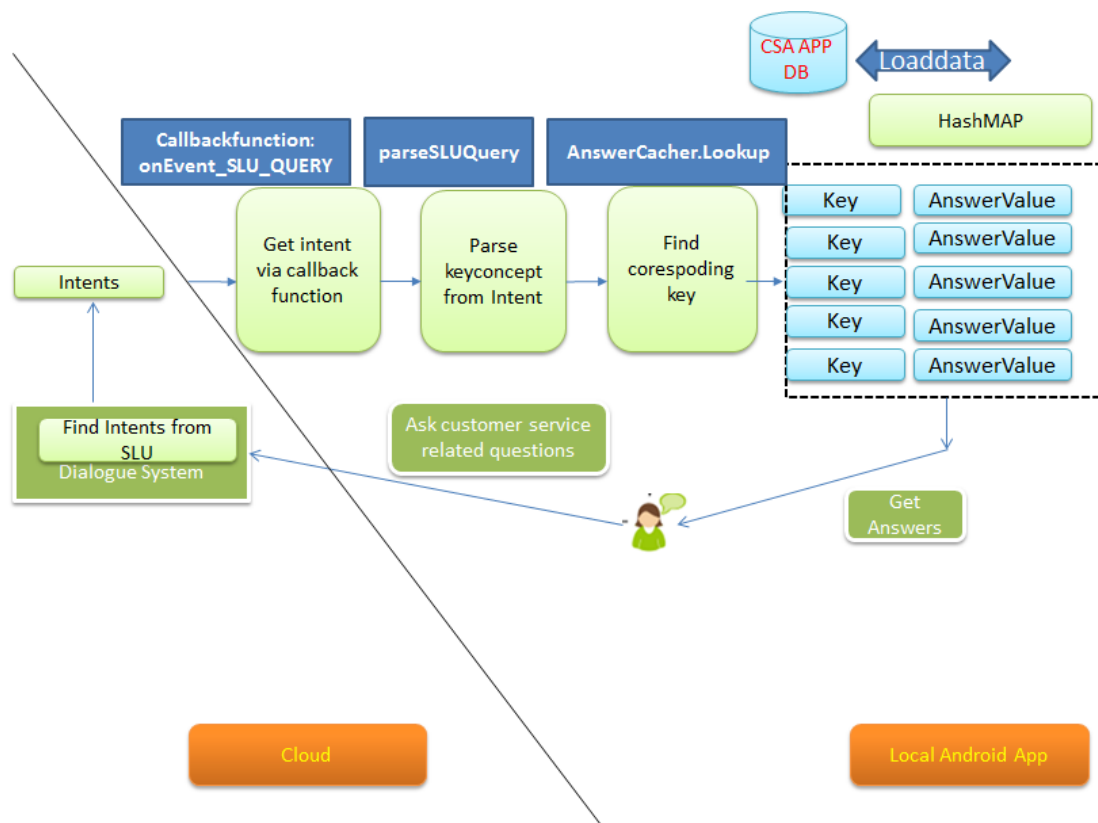


- 1) Android app to get voice signal or text from users
  - 2) Check if rule-based system get the intent/slots or not
  - 3) if yes, rule-based system handle it and response to user directly  
if not, call web API and return json(for example (`{'slot': ['o', 'o', 'S-Product.S-HowToKnow', 'S-Product.S-Price'],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

```

/*
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 voçonUtterance=mDSAPI_Result.event_slq_query.user_utterance_voçon();
    //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_slq_query.app_semantic();
    String error_code = mDSAPI_Result.event_slq_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 = cachex.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) {
    // TODO Auto-generated method stub
    //DSAPI_Result mDSAPI_Result = new DSAPI_Result(Result.toString());
    //String voçonUtterance=mDSAPI_Result.event_slq_query.user_utterance_voçon();
    //mTpmsHandler.sendMessage(mTpmsHandler.obtainMessage(MSG_onSluResult, Result.toString()));
    org.json.simple.JSONObject keyconcepts = new org.json.simple.JSONObject();
    String result = null;
    DSAPI_Result mDSAPI_Result = new DSAPI_Result(Result.toString());
    JSONObject app_semantic = mDSAPI_Result.event_slq_query.app_semantic();
    String error_code = mDSAPI_Result.event_slq_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 = cachex.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:

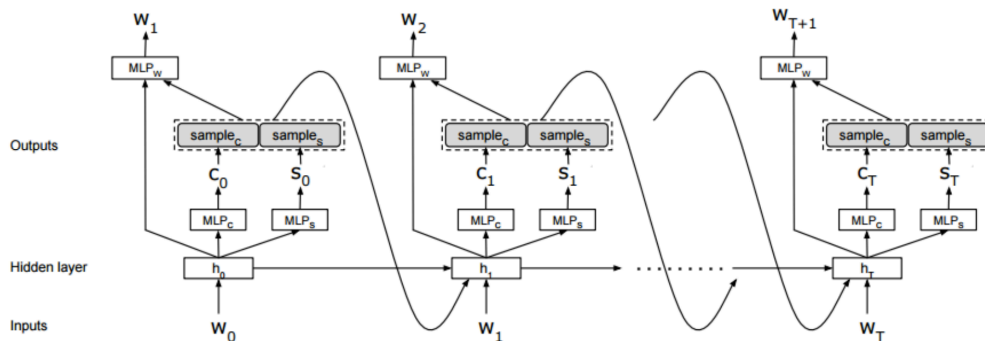
{Domain: Intentid, Slots }

## ◆ 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:

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

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'}

## 3.3 Program List

List functionalities of each file

### 3.3.1 AsusServiceMainActivity.java

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

### 3.3.2 AnswerCacher.java

- Extend AbstractGoldAnswerCacher class
- Implement lookup function

- Implement loaddata function

### 3.3.3 AbstractGoldAnswerCacher.java

- Implement lookup function
- Implement hash map using abstract class

## 3.4 API Interface

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