ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

SPOKEN INFOBOT DESIGN

M.Sc. THESIS

Ramazan GÖKAY

Department of Mechatronics Engineering

Mechatronics Engineering Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

SPOKEN INFOBOT DESIGN

M.Sc. THESIS

Ramazan GÖKAY
(518151023)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Asst. Prof. Dr. Hülya YALÇIN

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KONUŞAN BİLGİ BOTU TASARIMI

YÜKSEK LİSANS TEZİ

Ramazan GÖKAY
(518151023)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Dr. Öğret. Üyesi Hülya YALÇIN

HAZİRAN 2019

Ramazan GÖKAY, a M.Sc. student of ITU Graduate School of Science Engineering and Technology student ID 518151023 successfully defended the thesis entitled "SPOKEN INFOBOT DESIGN", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Dr. Hülya YALÇIN** Istanbul Technical University	....
Jury Members :	**Assoc. Prof. Dr. Gülay Öke GÜNEL** Istanbul Technical University	....
	Asst. Prof. Dr. Ebru Arısoy SARAÇLAR MEF University	....

Date of Submission : **3 May 2019**
Date of Defense : **13 June 2019**

To my family,

FOREWORD

I thank to my advisor Hülya Yalçın for her support throughout my thesis. I am grateful to Ahmet Afşin Akın and other TÜBİTAK BİLGEM Speech and Language Technologies Laboratory members for guidance and enduring my questions. Finally, I thank to Gökçe and my family for always being there for me.

June 2019 Ramazan GÖKAY
Electrical-Electronics Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

ASR	: Automatic Speech Recognition
CER	: Character Error Rate
CTC	: Connectionist Temporal Classification
DNN	: Deep Neural Network
DM	: Dialogue Management
DST	: Dialogue State Tracker
DSTC	: Dialogue State Tracking Challenge
GMM	: Gaussian Mixture Models
GOB	: Goal Oriented Bot
GRU	: Gated Recurrent Unit
HMM	: Hidden Markov Models
LSTM	: Long Short Term Memory
NLG	: Natural Language Generation
NLP	: Natural Language Processing
NLU	: Natural Language Understanding
PESQ	: Perceptual Evaluation of Speech Quality
RL	: Reinforcement Learning
RNN	: Recurrent Neural Network
SDS	: Spoken Dialogue System
TTS	: Text To Speech
WER	: Word Error Ratio

LIST OF TABLES

Page

LIST OF FIGURES

Page

SPOKEN INFOBOT DESIGN

SUMMARY

In recent years, significant amount of developments are obtained in technology. One of the developing technology field is speech and language technology. While machine learning and deep learning methodologies are used widely in speech and language technologies, human-machine interaction is carried on the new level. This human-machine interaction becomes more smooth and easy through new technologies.

The main purpose of this thesis is to design an infobot system for a specific domain so that it provides easiness and time saving to search and find information about a specific domain. In this thesis, restaurant searching domain is executed as specific domain.

Besides the main purpose of this thesis, this system should be user friendly and reconfigurable for new domains. To satisfy these requirements, speech interface and modular structure are embraced for this infobot system. Hence, spoken infobot system is designed in scope of this thesis.

This thesis consists of seven chapters. In the first chapter, the purpose of thesis, contribution of thesis and the outline of thesis are mentioned.

In second chapter, spoken dialogue system (SDS) is explained. General literature review is presented. The structure and components of spoken dialogue systems are mentioned briefly. Proposed SDS structure is introduced and explained why this structure is determined.

In chapter three, automatic speech recognition (ASR) which is the first module of the proposed spoken dialogue system is introduced. The logic behind the ASR problem and the components of ASR are shown. Applied ASR architecture is explained.

In fourth chapter, goal oriented bot (GOB) which is the second module and the brain of the proposed spoken dialogue system is introduced. The components of GOB are explained and the existing methods are mentioned. Applied GOB architecture and the working flow are presented.

In chapter five, text to speech (TTS) which is the last module of the proposed spoken dialogue system is introduced. Typical TTS pipeline is presented. The structure and the components of TTS system are explained. The existing TTS methods and examples are told. Applied TTS architecture is explained.

In chapter six, experimental studies done in the scope of this thesis are presented. Used datasets and hyperparameters are mentioned. Experiment results are discussed.

In seventh and the last chapter, all the work done in the thesis is summarized and concluded.

KONUŞAN BİLGİ BOTU TASARIMI

ÖZET

Günümüzde, yapay zeka ve makine öğrenmesi uygulamaları gittikçe önem kazanan bir hal almaktadır. Bu makine öğrenmesi uygulamaları çok farklı alanlar için kullanılabilir. Örneğin, veri madenciliği, robotik, sosyal medya analizi, doğal dil işleme gibi alanlar makine öğrenmesi tekniklerinin sıklıkla kullanıldığı alanlardır.

Gün geçtikçe daha da artan teknolojik yenilikler, elde edilen verinin ve hesaplama gücünün de artmasını sağlamıştır. Bu durum, makine öğrenmesi ve derin öğrenme çalışmalarına daha fazla dikkat çekilmesine neden olmuştur. Makine öğrenmesi ve derin öğrenme tekniklerinin konuşma ve dil teknolojilerine başarılı bir şekilde uyarlanmasıyla insan-makine etkileşimi yeni bir boyut kazanarak daha kolay ve akıcı bir hale gelmiştir. Önceki yıllarda, insanlar makinelerin nasıl çalıştığını anlamaya çalışıp görsel arayüz üzerinden yazarak ya da menülerde tıklayarak işlem yaptırıyorlardı. Günümüzde ise, bu işlemler sesli komutlarla yapılabilmektedir. Bu sesli komut uygulamaları daha da geliştirilip yaygınlaştırılmaktadır. Örneğin, Apple Siri, Google Now, Amazon Echo, Microsoft Cortana gibi sesli komutla çalışan uygulamalar bir çok kişinin hayatında bir şekilde yer edinmektedir. Gelişen teknolojiyle birlikte basit komut algılama uygulaması, daha karmaşık komutları bile algılama, anlamlandırma ve uygun cevap verme uygulamasına doğru evrimleşmektedir. Böylece insanlara, makinelerle sohbet etme veya diyalog kurabilme deneyimi sağlanmaya çalışılmaktadır.

Bu tez çalışmasının ana amacı kullanıcılar için bir bilgi botu tasarlamaktır. Böylece kullanıcıların, aradıklarıları bilgiye kısa zamanda ve kolaylıkla ulaşabilmeleri amaçlanmıştır. Ana amacının dışında, sistemin kullanıcı dostu ve farklı özellikler için yeniden yapılandırılabilir olması da amaçlanmıştır.

Kullanım kolaylığı sağlamak için, sesli arayüze sahip bir sistem düşünülmüştür. Çünkü insanların en temel iletişim yöntemi konuşmadır. Sesli komut ile işlem yaptırmak ve cevabı sesli olarak alabilmek kullanıcılar için büyük kolaylıktır. Örneğin, görme engelli veya felçli bir kullanıcı sistemi kullanmak için yazmak veya menü üzerinden seçmek zorunda kalmayacak. Benzer şekilde, teknolojiyle arası çok iyi olmayan veya okuma yazma bilmeyen kullanıcılar da bu sistemi kullanabileceklerdir. Kullanıcının tek yapması gereken konuşmak ve beklemek. Sistem konuşmayı anlayıp uygun cevabı sesli olarak döndürecektir. Bu tez kapsamında, restoran arama için konuşan bilgi botu geliştirilmiştir. Kullanıcılar, konum, yemek türü, fiyat aralığı bağlamında sesli olarak sorgu yapabilirler. Sistem, cevabı sesli olarak kullanıcıya döner.

Yeniden yapılandırılabilir olması için, modüler yapıya sahip bir sistem düşünüldü. Böylece sistem, ilerde istenebilecek özellik ekleme veya çıkarma yeteneğine sahip olmuştur. Sistem, piyasadaki ticari muadillerinden farklı olarak, açık kaynaklı bir yapıya sahiptir. Bütün sistem veya modüller farklı bir alan veya farklı bir dil için

yeniden eğitilebilir. Sistemin bir başka yeteneği ise çevrimdışı olarak çalışabilmesidir. Böylelikle bilgi güvenliği de sağlanmış olur.

Tez kapsamında geliştirilen, konuşan bilgi botu sistemi restoran arama amaçlı geliştirilmiştir. Bu sistem, bir çeşit konuşan diyalog sistemi olarak düşünülür. Tipik bir konuşan diyalog sistemi beş ana modülden oluşur: otomatik konuşma tanıma, doğal dil anlama, diyalog yönetimi, doğal dil üretimi ve konuşma sentezleme. Otomatik konuşma tanıma sistemi, konuşmayı yazıya çeviren bir sistemdir. Doğal dil anlama modülü, çevrilen yazıyı anlamlandıran bir modüldür. Diyalog yönetimi iki alt modülden oluşur:diyalog durum takibi, diyalog politikası. Diyalog durum takibi modülü, önceki diyalogların bilgisini tutan, yeni gelen söyleyişe göre eklenmesi veya güncellenmesi gereken bilgileri düzenleyen bir yapıdır. Diyalog politika modülü, anlamlandırılan bilgiler ve takibi tutulan durumlara göre uygun eylemi belirleyen bir modüldür. Doğal dil üretim modülü, kararlaştırılan eylemleri insanların anlayabileceği cümle formatına dönüştüren bir yapıdır. Konuşma sentezleme sistemi ise yazıyı sese dönüştüren bir sistemdir. Geleneksel diyalog sisteminde, bu modüller seri olarak birbiri ardına bağlantılıdır. Bu yöntem hem cevap süresinin hem de hatanın artmasına neden olur. Çünkü bir modül, kendisinden önceki modülün işini tamamlamasını beklemek zorunda ve önceki modüldeki hata sonraki modüle de aktarılır. Bu tezde, geleneksel sisteme alternatif olarak daha öz bir yöntem önerilmiştir. Ortada yer alan üç modül: doğal dil anlama, diyalog yönetimi ve doğal dil üretimi birleştirilerek amaç odaklı bot olarak görev alır. Önerilen konuşan diyalog sistemi üç ana modülden oluşur: otomatik konuşma tanıma, amaç odaklı bot ve konuşma sentezleme.

Konuşma tanıma, diyalog sisteminin ilk modülüdür. Bu modülün çıktısı diyalog sistemini beslediği için, bu modül diyalog sisteminin performansı açısından önemlidir. Bu tez kapsamında, konuşma tanıma modülü için Deep Speech 2 tabanlı bir mimari kullanılmıştır. Bu mimari, uçtan uca bir konuşma tanıma sistemidir ve yinelemeli sinir ağlarından oluşur.

Amaç odaklı bot ise diyalog sisteminin orta modülü ve aynı zamanda beynidir. Otomatik konuşma tanıma modülünden aldığı yazılandırılmış veriyi anlamlandırıp cevap üreten modüldür. Diyalog sistemi bu modül üzerine kuruludur. Bu tez kapsamında, amaç odaklı bot modülü için DeepPavlov tabanlı bir mimari kullanılmıştır. Bu mimari, uçtan uca bir mimari olup yinelemeli sinir ağlarını ve önceden belirlenmiş şablonları kullanan melez bir yapıdır.

Konuşma sentezleme, diyalog sisteminin son modülüdür. Amaç odaklı bot modülünden elde edilen cevaplar, konuşma sentezleme modülü ile seslendirilir ve kullanıcıya iletilir. Bu tez kapsamında, konuşma sentezleme modülü için Tacotron tabanlı uçtan uca bir konuşma sentezleme mimarisi kullanılmıştır.

Bütün bu modüller elde edildikten sonra birleştirilip ana sistem oluşturulmuştur. Bu tez kapsamında gerçeklenen deneysel çalışmalar sunulmuştur. Her bir modülün hangi veri kümesi ve hangi parametreler ile nasıl deneyler yapıldığı sunulmuştur. Elde edilen sonuçlar tartışılmıştır.

Otomatik konuşma tanıma modülü için, üç farklı veri kümesi ile deneysel çalışmalar yapılmış ve otomatik konuşma modülü farklı farklı bakış açılarıyla değerlendirilmiştir. Öncelikle, gerçeklenen konuşma tanıma algoritmasının testi yapılmıştır. Orjinal çalışmada kullanılan veri seti ile test edilmiş ve elde edilen sonuçlar, orjinal sonuçlar ile karşılaştırılmıştır. Bu karşılaştırmaya göre, gerçeklenen algoritmanın orjinali kadar iyi olmadığı fakat yine de kabul edilebilir seviyede olduğu çıkarımı yapılmıştır. Konuşma tanımayı etkileyen önemli bir faktör olan aksan meselesi için aksanlı konuşmalar içeren başka bir veri seti ile deneyler yapılmıştır. Sistemin gerçek kullanımında, farklı farklı aksanlı konuşmalar ile karşılaşalacağı göz önüne alındığından daha gerçekçi bir deney ortamı ele alınmış olur. Konuşma tanımanın önemli bir elemanı olan akustik modelin daha da iyileştirilebilmesi için de çalışmalar yapılmıştır. Eğitim verisinin veya model derinliğinin arttırılmasının konuşma tanıma üzerinde pozitif etkisi olduğu gözlenmiştir. Konuşma tanıma modülünün, gerçek kullanım senaryosunda karşılaşabileceği daha muhtemel olan bir veri seti ile de deneysel çalışmalar yapılmıştır. Bu bize sistemin gerçek kullanımında başarımı için daha gerçekçi bir sonuç göstermiştir. Gerçek kullanım senaryosuna daha yakın akustik veya içerikli veri ile sistem modellemek, performansı arttıracaktır. Eğer alana uygun yeterli veri elde yoksa, önce genel bir baz model eğitip daha sonra alan benzerliği yüksek bir veri ile aktarma öğrenimi benzeri bir eğitim yapılarak da sistemin başarımının arttırılabildiği gözlenmiştir.

Amaç odaklı bot modülü için, konum, yemek türü, fiyat aralığı bağlamında restoran sorgusu yapılabilen bir veri kümesi üzerinde deneysel çalışmalar yapılmıştır. Bu bağlamda diyalog sistemine girdi olarak gelen verinin anlamlandırılması ve diyalog sisteminin genel başarımı üzerine deneysel çalışmalar yapılmıştır. Bunun için niyet sınıflama, isimli varlık tanıma, boşluk doldurma ve amaç odaklı bot testleri yapılmıştır. Doğal dil anlama modülüyle ilişkilendirilebilen niyet sınıflama, isimli varlık tanıma, boşluk doldurma gibi testlerde yüksek başarım elde edilirken amaç odaklı bot testinin ise başarımın daha düşük olduğu gözlemlendi. Bu durumun, diyalog politikasının yeterince iyi modellenememesinden kaynaklanabileceği düşünülmektedir.

Bütün modüller birleştirildikten sonra, birleştirilmiş ana sistem için gerçek kullanım senaryosuna uygun olarak ampirik ve öznel testler yapılmıştır. Kullanıcı sesli olarak isteğini sisteme bildirmiş, sistem uygun cevabı üretip sesli olarak kullanıcıya geri dönmüştür. Sistemin ürettiği cevabın, kullanıcının isteğini ne kadar karşıladığı üzerinden bir test yapılmıştır. Öznel bir test uygulanmış oldu fakat yine de sistemin gerçek kullanım senaryosunda verebileceği tepkiyi görebilmek adına bir hissiyat oluşturmuştur.

Sonuç olarak, bu tezde amaçlandığı gibi, kullanıcı dostu, modüler ve tekrar ayarlanabilir bir bilgi botu sistemi geliştirilmiştir.

1. INTRODUCTION

In recent decades, the artificial intelligence and machine learning applications have become quite important in human life. Different machine learning tasks are used widely in many areas such as data mining, social media analysis, robotics, natural language processing (NLP).

While machine learning and deep learning methods are become widespread in speech and language technologies, human-machine interaction becomes more smooth and easy. In the past decades, people had tried to understand machines' working principles. On the other hand in recent decades, people try to make machines understand themselves. For example, people can control the machines by the voice through automatic speech recognition (ASR). ASR is the system that converts speech signals into text. For another example,the machines can talk like humans through text to speech (TTS) system. TTS converts text data to speech. These systems can be used for information services, chat-bots, dialog systems or smart assistants widely. Even cellphones have these technology. This situation leads to human-machine to get closer each other and reshapes the human-machine interaction.

1.1 Purpose of Thesis

The main objective of this thesis is to design an infobot system for a specific domain. This system can be used for various areas such as shopping-centers, hospitals, airports, schools etc. The system aims to provide easiness and time saving to search and find avaliable informations in specified field or location. In this scope, the subtask of the thesis is to make this infobot system user friendly and configurable as much as possible.

To provide user friendly purpose, it is decided that the infobot system should have speech interface, thus the system can communicate with user in human's most natural and general communication way. There is no need for learning new technology, all the user need is just speaking. Some people can have some motor-functional disease such as blindness, stroke etc. In this situation, these handicuped people has to deal with some problems to reach or use the conventional systems. In the conventional systems, the user must be close to the monitor or the user must write his/her searchings. However in proposed approach, this speech interface feature eliminates these conditions and provides some easiness for handicuped people. It is also good oppornutity for too old or too young people which are illiterate or not good at new technologies. They can reach their goals through just speaking easily.

To provide configurable structure, the system should have modular and retrainable structure. Within this configurable ability, the system can be extendable for new features or it can be adapted to new domains and new languages.

To conclude that, the aim of this thesis evolves to design modular spoken infobot system.

1.2 Contribution of Thesis

In this thesis, a spoken infobot system is developed. This infobot includes some important modules such as speech recognition, natural language understanding, dialogue management, natural language generation and speech synthesis. Each of these modules is a hot research topic all by itself. This infobot system combines all these modules and uses for people's benefits.

This developed infobot is not black-box unlike some commercially well-known products. This infobot is open source, modular, reconfigurable and retrainable. The user can modify the system as he wants. Another ability of this system is to work offline. The system does not need internet connection, all the processes are completed in locally. Hence it provides information security.

To summarize the contributions of the thesis; the output of this thesis provides easiness for end-user and it also provides some research opportunities to developers for recent speech and language technologies. In scope of this thesis, an useful infobot system which can be addressed different type of purposes is developed.

In scope of thesis studies, obtained main outcomes are listed as follows:

- English Spoken Infobot system for restaurant searching
- English ASR system

- English TTS system
- English GOB system
- Turkish ASR system
- Turkish TTS system
- A conference paper
  - **Gokay R.**, Yalcin H., Improving Low Resource Turkish Speech Recognition with Data Augmentation and TTS. *16th IEEE International Multi-Conference on Systems, Signals and Devices 2019 (SSD 2019)*, 21-24 March 2019, Istanbul, Turkey.

1.3 Outline of Thesis

The remaining of thesis is organised as follows.

Chapter 2 gives some information about spoken dialogue system and the components of dialogue systems.

Chapter 3 mentions about automatic speech recognition (ASR) and presents our approach for this thesis.

Chapter 4 presents goal oriented bot (GOB) structure and the how to applied in our thesis.

Chapter 5 talks about text to speech (TTS) system and how to applied in the thesis.

Chapter 6 presents some experimental studies and discusses the results.

Chapter 7 summaries the studies and concludes the thesis.

2. SPOKEN DIALOGUE SYSTEMS

Technological developments on speech and language processing fields reshape the human-machine interaction style. In the past years, the user had to provide input for the machine by typing text or clicking menu item in the graphical interface. However, in the recent years, the user can control the machine by voice and get voice response. Voice controlled systems have become familiar to human life day by day. For example personal assistants like Apple's Siri, Amazon's Echo, Google Now and Microsoft's Cortana perform voice-controlled functions such as calling a person, internet surfing or getting weather information.

The underlying purpose of spoken dialogue systems is to provide human-like style communication ability to computers. The motivation for this purpose comes from communication easiness. People have the speech communication ability naturally, so that they do not perform extra effort. Thus, spoken dialogue systems offers to humans natural and effortless interactive experiences with machine.

2.1 Literature Study

Dialogue systems can be used in various applications such as smart assistants, technical support services, chatting services. Based on usage purposes, dialogue systems can be distincted into two class: goal-oriented, non goal-oriented. Although both types of dialogue systems have some objectives, the goal-oriented dialogue systems have more specific purposes. That's why, they take more place in human life compared to non-goal-oriented systems. The boundary uncertainity of the non-goal-oriented system makes it less preferable. Another issue is the data requirement to train the system. Non-goal-oriented systems require enormous data compared to goal-oriented systems.

This thesis aims to design a spoken infobot for the restaurant searching based on location, cuisine type or price range. The purpose is specific and well-defined, so that designing a goal-oriented dialogue system is aimed in this thesis. Therefore the focus is headed to goal-oriented dialogue system studies.

In goal-oriented dialogue systems, initial works consists of rule-based techniques [5]. Later, machine learning methods are used for the intent classification and dialogue systems. In [6], a telephone based bus schedule information system which is called as Let's Go is developed as task-oriented spoken dialogue system. In another study [7], a speech enabled kiosk is developed to guide students or visitors through their institute. In that study, latent semantic analysis statistical method is used for natural language understanding.

In [8], the dialogue system is formulated as a sequential decision making problem based on Markov decision processes. In [9], a statistical dialogue system is developed for public transport information domain. That study provides various automatic speech recognition and natural language understanding options. In more recent studies, deep learning methods take place of statistical methods. Indeed, end to end dialogue systems are developed instead of multi-modular structure. In [10], an end to end dialogue system is developed by using memory networks.

2.2 Spoken Dialogue System Structure

Typical spoken dialogue systems consist of five main modules: automatic speech recognition, natural language understanding, dialogue management, natural language generation and speech synthesis. In a conventional configuration these modules are connected in serially, as shown in Figure 2.1. The general workflow for this configuration is as follows:

- Firstly the user utterance is converted to text by using automatic speech recognition (ASR) module.
- The converted text is passed to natural language understanding (NLU) module to interpret the user intent.
- The dialogue management (DM) module decides the proper action. This action can be database query or requesting extra information.
- Decided actions are converted to human understandable sentence format by natural language generetion (NLG) module
- Finally, this sentence is vocalized by text-to-speech (TTS) module.

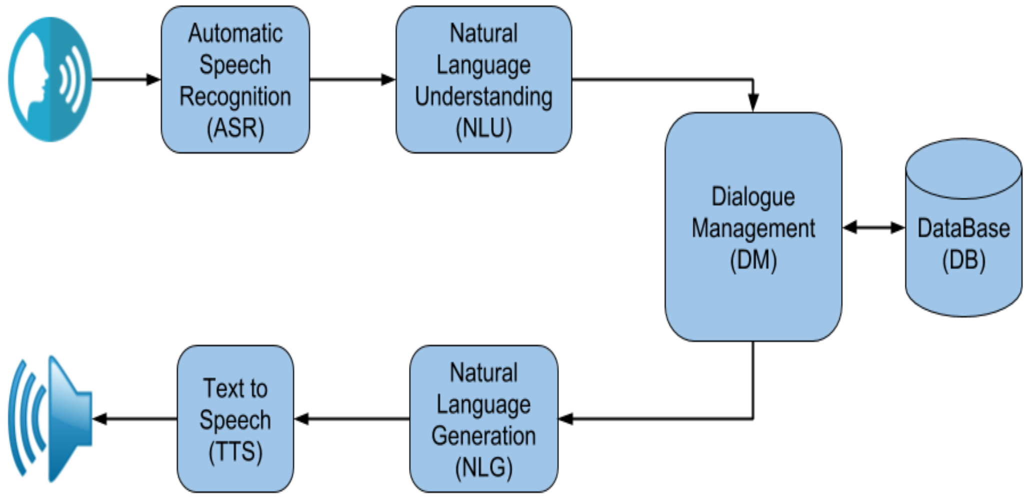

Figure 2.1 : Spoken dialogue system structure

A dialogue system generally is built on some sort of databases, so that the system can query for the user intent and then it returns the output of database query to the user.

In pipeline dialogue system, the information is passed from previous module to next module. This situation makes the dialogue flow simple, however it causes some drawbacks. One module has to wait until the previous module completes its job. This waitings cause the time delays. Another issue is the error propagation problem. Each module has some error range, this error is passed through modules in pipeline dialogue system. Thus the error propagates through modules.

An alternative solution to the pipeline architecture is the blackboard architecture [11]. In this architecture, the information exchange between components is provided a shared space [12] as shown in Figure 2.2.

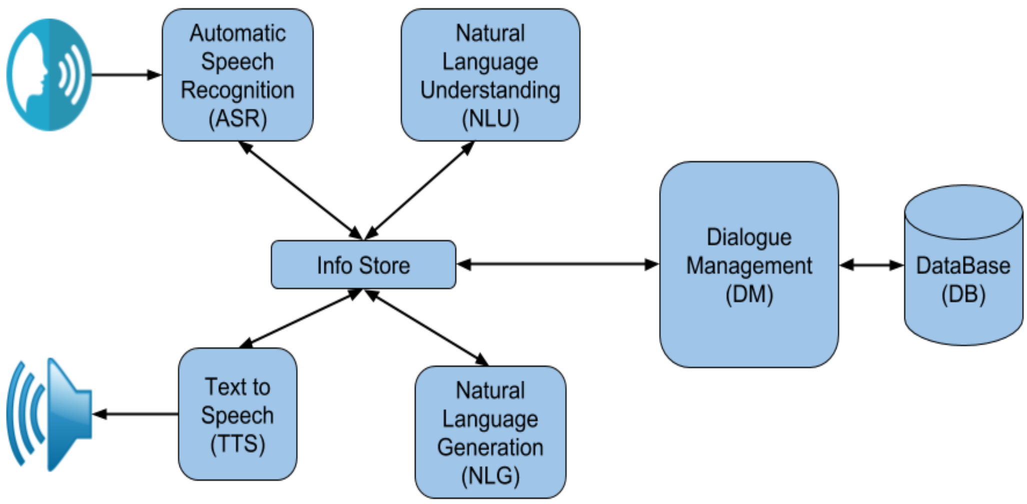

Figure 2.2 : Spoken dialogue system blackboard structure

2.2.1 Automatic speech recognition (ASR)

Automatic speech recognition (ASR) system converts speech utterances into word sequences. To do that, the speech signal is separated from silence regions by using voice activity detection (VAD). A simple approach for VAD is based on energy level comparison. The region which has lower energy level compared to some predetermined threshold represents silence and the rest represents the speech segments. However, this VAD task may not be as easy as this situation. For example, the background noise in the environment can cause problem to detect speech segments. For this reason, more advanced approaches may be taken into account to distinguish speech, silence and noise. After the speech segment is obtained, the speech segment is decoded to get the corresponding sequence of words.

ASR system consists of two parts: training and decoding. In training part, some acoustical patterns of characters and some linguistic patterns of words are derived. In decoding phase, the input speech segment is matched with derived acoustical and linguistic patterns. By adding pronuncation dictionary to that matching, the most probable word sequence is decoded.

2.2.2 Natural language understanding (NLU)

The ASR system output hypothesis is passed to natural language understanding (NLU) module. NLU produces the semantic represantations of the ASR hypothesis. That means to convert the ASR hypothesis to understandable format for the system. This includes a task of parsing, slot filling and intent detection. For example, the intent could be a *"greeting"* like *"Hello"*, or it clould have an *"request"* nature, for example *"I am looking for a Chinese restaurant"*. Depending on the purposes, the slots could be various, like the location, price, start time, name, year etc. The intents and slots represents the characteristics of the dialog system. These slot filling and intent detection task is behaved as a sequence tagging problem. That's why, the NLU can be implemented as a recurrent neural network architecture. For example, it can be implemented as a Long Short Term Memory (LSTM) with Conditional Random Field (CRF) approach.

To achieve slot filling and intent tasks, the dataset includes concatenated B–I–O (Begin,Inside, Outside) slot tags, the intent tag and an additional end-of-string (EOS) tag. As an example, in a restaurant reservation case, given the sentence *"Is there any Indian restaurant in Toronto downtown?"*, the task is to correctly output, or fill, the following slots: *{cuisine: Indian}* and *{location: Toronto downtown}*. The Table 2.1 shows the tagging of the previous example.

Table 2.1 : Tagging sentence example in B-I-O format.

Is	there	any	Indian	restaurant	in	Toronto	downtown?
O	O	O	B-Cuisine	O	O	B-Location	I-Location

Through NLU module, the domain of user utterance is detected, the user intent is classified and the intent entities are recognized. The output of the NLU module is feed to Dialogue Management module.

2.2.3 Dialogue management (DM)

Dialogue management (DM) is the brain of the spoken dialogue systems. It coordinates the flow of dialogue. Dialogue Manager creates a link between user utterance and the system response. Also, DM could be connected to some external Knowledge Base (KB) or Data Base (DB) structures to provide the right response.

DM makes the decisions beased on the information comes from NLU module and then decides what to do on the background of this information. When the system is more complex than a simple question-answering system, the DM has to keep track of the dialogue such as which information has been understood from coming dialog acts, what is uncertain and the system needs verification and what the system still needs to know etc. Dialogue manager includes two components which are dialogue state tracking (DST) and policy learning (PL).

Dialogue state tracker

The job of the dialogue state tracker module is to produce state s_t for dialogue segment in time t. The s_t is a structure that provides all necessary information for the agent. According to s_t information, the agent could apply the next action.

DST takes the observable inputs such as all previous user utterances, the system responses, all the NLU outputs and all external informations provided by some database module. For instance, in a restaurant search case, the state might indicate the price range and cuisine preferences of the user or what information he is looking for like a location, telephone number etc.

According to given all of this information DST results the probability distribution over all possible dialogue states [2]. The true state can not be observed from the raw input because of some factors such as unclear specified user utterances, the noise, the error of NLU or the changes in user goal [13]. It is also important that the input of DM should be as rich featuare as possible in order to make right decisions about the dialogue state and user intention [14], [15].

Policy learning (PL)

The policy learning module selects the next system actions to drive the user towards the goal in the smallest number of steps [2]. The state representation s_t includes the user actions, system actions, avaliable database results, turn information and history of dialogue. According to state representation s_t, the policy π generates the next system action with respect to $\pi(s_t)$ function. The function can be optimized by using supervised learning or reinforcement learning [16].

2.2.4 Natural language generation (NLG)

Natural language generation (NLG) can be explained as the opposite of the NLU. DM's system response is mapped to understandable format for the end user by NLG. Depending on the task, various NLG approaches could be applied. As a basic approach, an answer can be selected a set of use-ready text messages. Another approach is rule-based approach. In rule-based approach, some predefined templates is used. Slot-filling mechanism can be applied in rule-based approach as well. These templates are generally task-specific, there is not much generalization power in rule-based approach. Even so, it can be modified for similar taskes easily. More advance approach is model-based NLG method. Model-based NLG modules have learnable parameters and are usually trained in a sequence-to-sequence approach. The models presented in [17] and [18] use an LSTM-decoder with a given semantic frame, to generate template-like sentences with slot placeholders. After that, a beam search is applied to replace the placeholders with actual values [2].

2.2.5 Text to speech (TTS)

The last module of the dialogue system is text to speech (TTS). The output of the NLG module is converted to speech through this module. This process involves two steps which are text-normalization and voice generation. In text-normalization step, all numeric characters are converted to word representations, abbreviations replaced with their long formats and the all punctuations are removed from the text. After that, each word is assigned to phonetic representation. This representation is sent to speech synthesis. There are various synthesis methods such as unit selection synthesis, statistical synthesis [11] and more recently neural synthesis methods. Speech is the most natural communication way of the people. That's why, the voice response is embraced by user easily and speech synthesis module improves the dialogue system experience.

2.3 Proposed Dialogue System Structure

A typical spoken dialogue system (SDS) includes five main modules: automatic speech recognition (ASR), natural language understanding (NLU), dialogue management (DM), natural language generation (NLG) and text to speech (TTS).

Pipeline SDS architecture is easy to apply, however it is in trouble through time consumption and error propogation problems. Each module has to wait until the previous module completes its job and each module has an error range. While the number of modules increases in dialogue system, total response time and the propogated error also increases.

In alternative solution blackboard SDS or incremental SDS, there is info-store module. Each module shares its output and executes simultaneously. The results are updated in each iteration. This alternative solution can handle with time consumption problem, however the system implementation is more complex. The number of modules is also same as typical pipeline SDS architecture, so that it can not solve the error propogation problem properly.

In proposed solution, text based parts of dialogue system is combined. End to end dialogue system approach is applied. Hence, new SDS solution includes three main modules: ASR, goal-oriented bot (GOB) and TTS as shown in Figure 2.3.

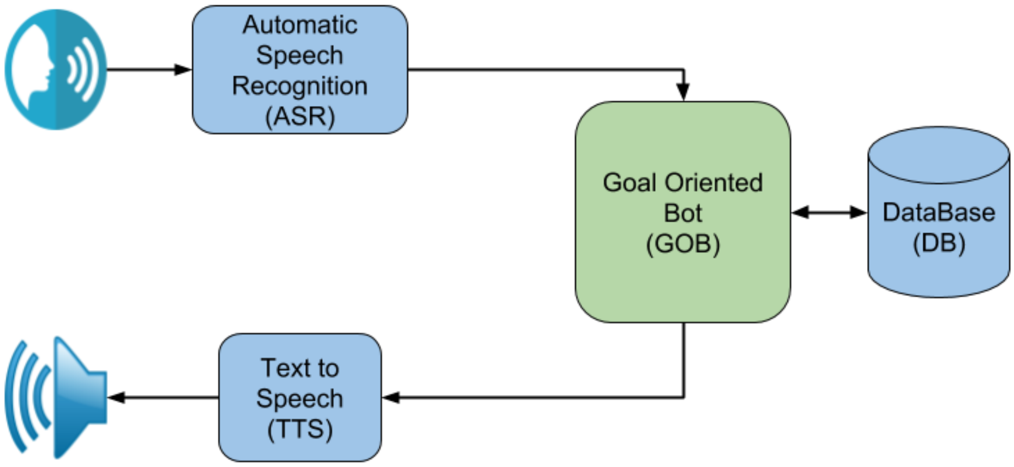

Figure 2.3 : Proposed spoken dialogue system structure

In typical pipeline architecture, these dialogue system modules have been built as a pipeline through language understanding, state tracking, action selection, and language generation modules. However, dependencies between these modules come with considerable amount of complexity. For instance, it can be unclear how to define the dialogue state and what history to maintain, yet action selection relies exclusively on the state for input. Moreover, training each module requires extra specialized labels [19]. GOB structure makes the job of NLU, DM and NLG modules as shown in Figure 2.4. It takes user utterance as text, the system response returns the text.

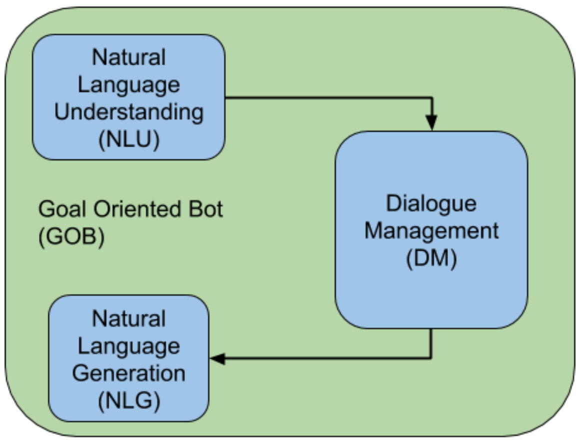

Figure 2.4 : Goal oriented bot structure

In scope of thesis studies, a modular spoken infobot system has been developed. To do that, three different module has been developed and these modules have been combined on common platform.

3. AUTOMATIC SPEECH RECOGNITION (ASR)

Automatic Speech Recognition (ASR) is a system that maps user utterances into sequence of words. The history of the speech recognition systems goes back to 1950s-1960s. The first speech recognition systems focused on just numbers. After some time, the systems also tried to focus on words. Initially, the researches were conducted for limited commands after that, researches was conducted on continious speech recognition. Through developing technology, the speech recognition systems also develop and show hopeful performance.

ASR problem is stated to find the most probable word sequence among all possible word sequences with respect to input speech utterance [20]. The utterances composed of words. The phone is minimal unit that distinguishes one word from another. The general problem is formulated for observable feature sequence $O = o_1, o_2, ...o_n$ and word sequnce $W = w1, w_2, ..w_m$ as shown

$$\hat{W} = argmax\ P(W|O) \tag{3.1}$$

By using Bayes' formula, equation (3.1) can be rewritten as

$$\hat{W} = argmax\ \frac{P(O|W) * P(W)}{P(O)} \tag{3.2}$$

where *P(O|W)* is the probability of the observable feature vector *O* given the speech utterance word sequence *W*. The *P(W)* is the probability of the saying word sequence *W*. The *P(O)* represents the probability of *O*. The *P(O)* does not change for each sentence, because same acoustic observation is examined for each potential sentence [15]. That's why it can be omitted and the equation (3.2) can be simplified as

$$\hat{W} = argmax\ P(O|W) * P(W) \tag{3.3}$$

In equation 3.3, log-likelyhood probability *P(O|W)* comes from acoustic features, and the prior probability *P(W)* comes from linguistic features.

3.1 ASR Structure

When the general ASR equation is rewritten for transcript W, phone p and speech sample O as follows

$$\hat{W} = argmax\ P(W|O) = argmax\ P(O|p) * P(p|W) * P(W) \tag{3.4}$$

In equation (3.4), *P(O|p)* represents *Acoustic Model*, *P(p|W)* represents *Lexicon* and *P(W)* represents *Language Model*. Typical ASR pipeline is shown in Figure 3.1.

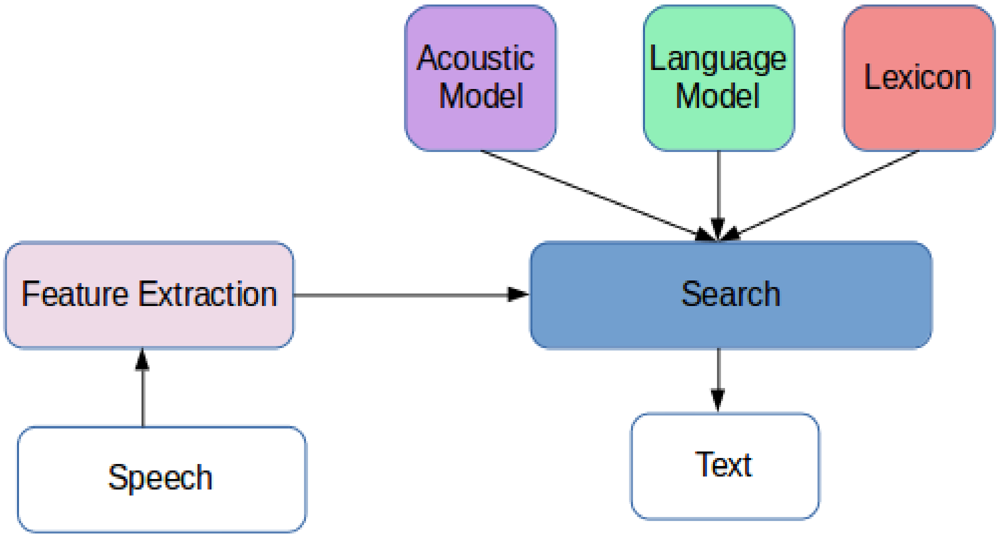

Figure 3.1 : Automatic speech recognition pipeline

3.1.1 Acoustic model

Acoustic model is mapping the speech signal to the discrete sounds of the words. These discrete sounds are called as phonemes. For example, there are 44 phonemes in English. All words can be represented by using these phonemes. The pronuncation of *"CAKE"* word is represented with phoneme string *"K EY K"*. The acoustic model learns to map phoneme string *"K EY K"* with corresponding speech signal.

To train acoustic model, large amount of speech data and exact transcription of the speech data are required. The speech signal is chunked with 20-30ms duration window within 10ms shifting. Each window is called as frame. Feature exraction is applied for each frame. Generally, mel-frequency cepstral coefficients (MFCC) feature is used for feature extraction stage. After that, first and second order derivatives of the features are computed and concatenated to original one so that acoustic feature vector is obtained.

Acoustic feature sequence is modelled by using Hidden Markov Models(HMM). In traditional systems, each phoneme is modelled by a separate HMM and the emission distribution of HMM states are modelled by Gaussian Mixture Models(GMM). After the spread of Deep Neural Networks (DNN), DNNs are used to produce posterior probabilities over HMM states.

3.1.2 Language model

Language model can be defined as set of word probabilities in a language corpus. Generally *n*-gram model is used for language modelling. In this model *n* represents sequential word number. For example, uni-gram represents single word, bi-gram represents two words, tri-gram represents three words. Language model gives prior probability of the word in word strings. Language model is generated from large ASR text data.

3.1.3 Lexicon

Lexicon is a pronunciation dictionary that maps each word to phone sequences. Usually, a pre-defined dictionary is used for ASR. However, this dictionary can be extended for adding new word or adding new pronuncation of existing word. A word can have different pronuncations. For example, Lexicon maps the word *"CAKE"* with phoneme string *"K EY K"*

3.2 Speech Recognition Methods

In this part, existing speech recognition methods are mentioned briefly. Firstly, statistical methods which are based on HMM and GMM are introduced. Next, deep learning based speech recognition methods are mentioned. Finally, end to end speech recognition methods are explained.

3.2.1 GMM/HMM based speech recognition

Typical ASR system has three main components: acoustic model, language model and lexicon as shown in Figure 3.1. In statistical methods, acoustic model is trained based on HMM and GMM.

Speech signal is parameterized and represented as an observation sequence $O = \{o_t\}$ and the ASR purpose is to find most probable word sequence $\hat{W}$ as shown in Figure 3.2.

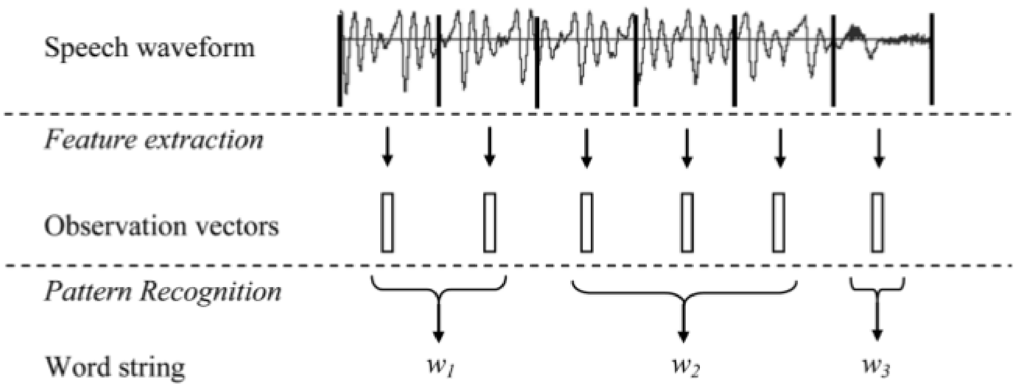

Figure 3.2 : Illustration of the speech recognition process [1]

The acoustic model can be represented as follows:

$$P(O|W) = \sum P(O|c) * P(c|p) * P(p|W) \tag{3.5}$$

In acoustic modelling, the probability distribution of the acoustic features for each state is determined by using GMM as shown in equation (3.6). In that equation, r, μ, σ symbols represent weight, mean and variance parameters respectively for entering state j, previous state i. The GMM is trained by using E-M algorithm [21].

$$P(o_t|c_i) = \sum r_{ij} N(o_t; \mu_{ij}, \sigma_{ij}) \tag{3.6}$$

The Word sequence is modelled within language model according to equation (3.7).

$$P(w_1, w_2, ..., w_N) = P(w_0) \prod P(w_i|w_0, ..., w_{i-1}) \tag{3.7}$$

In the decoding phase, the task is to find the most likely word sequence W given the observation sequence O according to trained acoustic model,language model and the lexicon. A searching algorithm is applied on the possible word sequences. Generally, Viterbi algorithm [22] is used to estimate most probable state sequence for the observation sequence. In the large vocabulary recognition tasks, it would be challenging to consider all possible words during the recursive part of the Viterbi algorithm [23]. To handle that situation, a beam search can be used for Viterbi algorithm, so that only the words with sequence probabilities above a threshold are considered.

3.2.2 DNN/HMM speech recognition

In traditional ASR systems, GMM/HMM paradigm is used mostly. However, GMM has a problem to model the data present on the boundary line [24]. Neural networks are more successful to model the data on the boundary condition. DNN based acoustic modelling improves the ASR performance. In first implementations, DNN is used for phoneme recognition and HMM is used for decoding. Hence, DNN/HMM paradigm is developed. Since DNNs have many layers and can produce many outputs, DNNs satisfy the large number of HMM states.

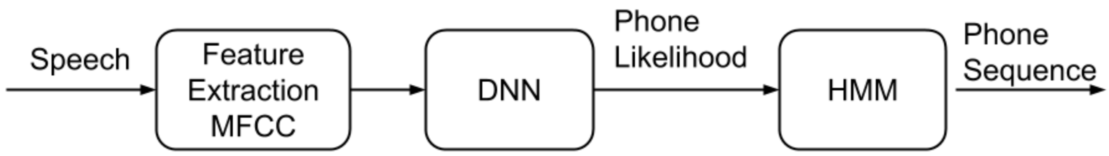

Figure 3.3 : DNN/HMM ASR structure

In first times, DNN is just used to get posterior probabilities of phonomes and the decoding process is applied by using HMM paradigm. The devlopments in deep neural network studie encourages the researchers to use DNNs for ASR completely.

3.2.3 End to end speech recognition

Conventional ASR systems include acoustic model, language model and pronuncation model components which are trained separately. Each component requires some sort of expertise and extra time to obtain proper formatted train data. This situation is time-consuming.

As alternative solution to conventional ASR system, end to end (E2E) ASR system is proposed. End to end ASR system directly maps a sequence of input acoustic features to sequnce of phonemes or words. Another advantage of the end to end ASR system is that it is trained according to final evaluation metric so that trained model is more optimized. The structural difference between typical ASR and end to end ASR can be seen in Figure 3.4.

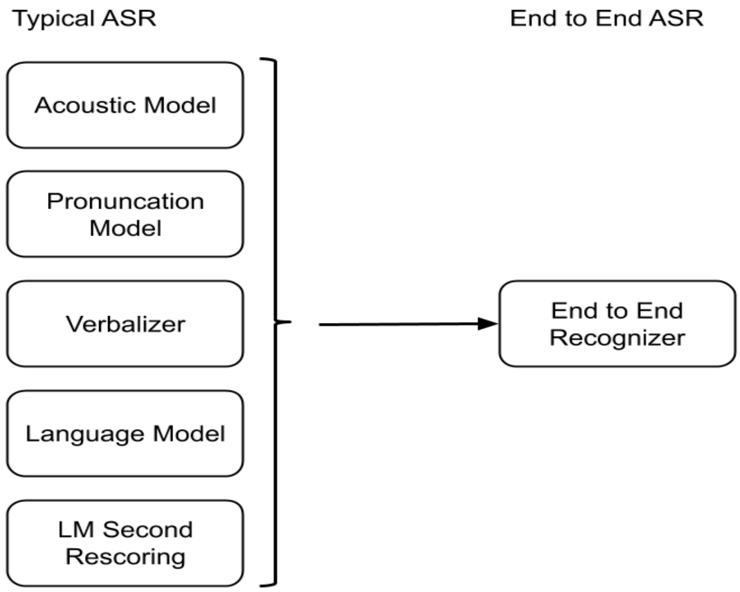

Figure 3.4 : Typical and E2E ASR structure

In [25], end to end ASR is applied by replacing speech recognition pipeline with single recurrent neural network(RNN) architecture. The spectograms are processed by using bidirectional Long Short Term Memory (biLSTM) with Connectionist Temporal Classification (CTC) [26] output layer. In that study, the model is trained directly on text transcriptions, so that there is no need for pronuncation dictionary. Besides that, there is no need for force alignment since CTC integrates out over all possible input-output sequence alignments. And the loss function trains the network to directly optimise for final evaluation metric which is the word error rate in ASR systems.

Connectionist temporal classification (CTC)

In speech recognition task, neural networks are trained as frame level for acoustic modelling. This requires alignment between audio and corresponding transcript. This is a challenging problem, since the input speech and output transcript lengths are different mostly. For example, a person speaks fastly and another person can speak slowly for the same word. Their transcription lengths are same, but their speech lenghts are different. In that situation, Connectionist Temporal Classification (CTC) [26] can solve that alignment problem by introducing character repetition and blank label indice. CTC gives the probability of a label for a given input, by summing over all the probabilities of the character for each time-step.

The CTC working princinle is illustrated for label sequence *{h,e,l,o,-}* in Table 3.1. General approach is as follows:

Table 3.1 : Example of CTC principle.

<table>
<tr><td>input</td><td>h</td><td>e</td><td>e</td><td>-</td><td>l</td><td>l</td><td>-</td><td>l</td><td>o</td><td>o</td></tr>
<tr><td>merge repeat</td><td>h</td><td colspan="2">e</td><td>-</td><td colspan="2">l</td><td>-</td><td>l</td><td colspan="2">o</td></tr>
<tr><td>remove blank</td><td>h</td><td colspan="2">e</td><td></td><td colspan="2">l</td><td></td><td>l</td><td colspan="2">o</td></tr>
<tr><td>output</td><td>h</td><td>e</td><td>l</td><td>l</td><td>o</td><td colspan="5"></td></tr>
</table>

- The CTC network assigns the highest probable characters or the blank token for input sequence
- Repeated characters are merged
- Blank tokens are removed
- Finally, output sequence is obtained

3.3 Applied ASR System

In this thesis, end to end ASR system approach is embraced instead of conventional ASR approach. Conventional systems requires extra efort and time to prepare some pronunciation or language specific needs. And these needs could require expert knowledge, so that it causes extra work loads. On the other hand, end to end ASR systems can eliminate these extra works.

Deep Speech 2 architecture [27] based system is applied as end to end ASR system in this study. Deep Speech 2 (DS2) is able to handle noisy environments, different accents and languages. There is no need for hand engineered components in DS2-based ASR.

The architecture of used DS2 system consists of 2 pieces of 2D invariant convolutional layers, 7 bidirectional Gated Recurrent Unit (GRU) layers and 1 fully connected layer as shown in Figure 3.5 and each GRU layer includes 768 hidden units. The ASR system was trained by using Connectionist Temporal Classification (CTC) loss function [26].

The applied ASR pipeline is summarized based on Figure 3.5 as follows:

- The audio spectogram is passed as input.
- The input is fed to neural networks.

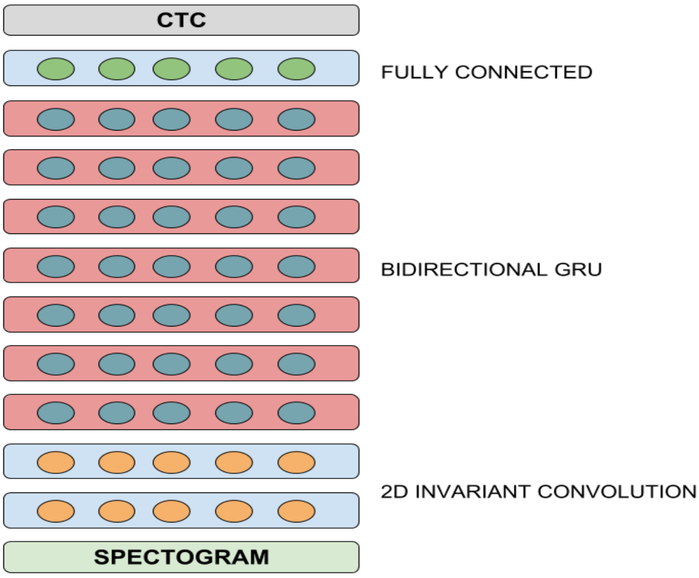

Figure 3.5 : Applied ASR structure

- The network gives conditional probability over outputs (letters and blank character) for each input step.
- The propabilities of different sequences are computed within per time-step output distribution.
- By marginalizing over alignments, the distribution over outputs is obtained.

During DS2 training, the network also learns implicit language model beside of acoustic model. In ASR inference time, the decoding can be executed by bare model so that it is called Greedy Search. However, that implicit language model is not enough if the amount of training data is small. In that situaion, an explicit language model can be coupled with DS2 model so that is called as Beam Search. In this thesis, 3-gram language model was generated by using KenLM toolkit [28]. This 3-gram language model was used in beam search decoding with respect to equation (3.8).

$$Q(y) = log(p_{ctc}(y|x)) + \alpha log(p_{lm}(y)) + \beta word\_count(y) \tag{3.8}$$

According to equation (3.8), the aim is to find transcription y which maximizes $Q(y)$. The coeffcient α controls the relative contributions of the language model and the CTC network [27]. The coefficient β controls the number of words in the transcription. α and β parameters are fine-tuned to get optimal transcription.

In scope of ASR studies in the thesis, various acoustic and language models have been generated. Many ASR experiments have been conducted to get better ASR system.

4. GOAL ORIENTED BOT (GOB)

Goal Oriented Bot (GOB) is a kind of closed domain text-based dialogue system that takes text as input and returns proper text response as output. This module is the brain of the spoken dialogue systems (SDS). In SDS, the output of ASR is passed through GOB module. The transcribed text data is interpreted by using syntactic and semantic analysises. Next, the corresponding actions are determined through dialogue management. After that, the system response is generated in human understandable sentence format. All of these text-interperetion, decision making, and sentence generation processes are executed by GOB module. Finally, the output of GOB module is vocalized by TTS module.

4.1 GOB Structure

A typical GOB structure is consisted of three components: natural language understanding, dialogue management, natural language generation. The general pipeline of typical GOB system is shown in Figure 4.1.

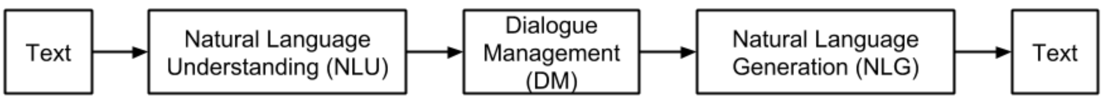

Figure 4.1 : Typical GOB structure

The first part of the GOB is the natural language understanding (NLU) module. It takes the transcribed utterance of the user and then returns the user intent and proper slots(input pieces). For example, when the input text *"I am looking for a chinese restaurant in the south of town"* is processed by NLU unit. NLU unit can return *"intent: resturant reservation"* and *"cuisine: chinese", "location: south of town"* as output. NLU has a stateless structure, since it does not log previous conversations. It is responsible with just intent detection and slot filling.

Second part of the GOB is the dialogue management (DM) module. The outputs of NLU which are the intent and new slots are passed to DM module. DM consists of two submodules: Dialogue State Tracking (DST) and Dialogue Policy (DP). DM has the previous conversation informations so that it has state of the conversations because of DST submodule. DST tracks the state of conversation by adding new one or changing the existing one. For example, when the new input text *"I am looking for a indian restaurant"* the state is updated from *"cuisine: chinese"* to *"cuisine: indian"* by DST. For another instance, when the new input text *"I want cheap restaurant"* the new slot is added as *"price-range: cheap"* by DST. Once the state of conversation is the updated, DP makes the decison about what to do next. It can ask follow up question to fill to new slot or it returns the proper answers from some kind of database based on current conversation state.

The final part of the GOB is the natural language generation (NLG). The output of the DM module is passed to NLG module. Hence, the system response is formatted to human understable style. NLG can use the template-base approaches as the output. For example, the template *"#name is a #price-range #cuisine restaurant in #location"* can be used for previous example. The slots *"#name, #price-range, #cuisine, #location"* are filled according to conversation states and the database query results.

4.2 GOB Methods

In this subsection, general task oriented dialogue system approaches are explained briefly. Each method has some kind of advantage and disadvantage in itself. Details are mentioned in following parts.

4.2.1 Supervised learning methods

Dialogue systems can be considered as some kind of machine translation problem. For example, in dialogue systems, the input is in text format, the output is in the text format similar to machine translation. Sequence to sequence modelling approach can be applied to dialogue system as well. That's why, some deep learning approaches which give hopeful results in machine translation field can also be used for dialogue systems. Deep neural networks based on recurrent neural networks (RNN) and encoder-decoder approaches can be used to predict latent dialogue states [2]. However, these approaches are pretty data-hungry, so that large amount of data is required.

In [29], standard RNNs are used and the dialog training is conducted sequence to sequence approach for IT helpdesk GOB purpose. The dataset consists of converstaions about IT troubleshooting and the probable solutions. The trained model tries to predict the next sentence based on the previous sentence or conversation. The system has end to end structure and requires less hand-crafted rules. However, recurrent neural nets have memory problems about reasoning-effect relation through long dependencies.

In [10], LSTMs are used instead of standard RNNs. LSTM with explicit memory combination gives better results compared to standard RNNs. That study is conducted by using bAbI tasks [30] for restaurant reservation domain. Dialogue State Tracking Challenge 2 (DSTC2) dataset [31] is used in that study. The system is more successful about remembering and updating the user preferences, so that it shows better performance about reasoning-effect relationship in conversation.

4.2.2 Reinforcement learning methods

In another perspective, the dialogue system can be modelled as a Partially Observable Markov Decision Process (POMDP) [8]. In that case, reinforcement learning (RL) [32] can be a good alternative solution for dialogue systems.

In that approach, typical dialogue flow is preserved. It still includes NLU, DST and NLG modules as shown in Figure 4.2. On the other hand, Dialogue Policy(DP) part which is reponsible for action selection is built on reinforcement learning (RL) approach.

In [33], a simple question-answering system is developed by using Reinforcement Learning. It answers the user questions about movies, actor and released year. In another study [16], a movie booking dialogue system is built based on RL approach. It helps the users about movie booking and seeking. Both studies include user simulator [34] to train the system since reinforcement learning approach requires interaction between dialogue agent and the environment but the conventional dialogue datasets can not be used directly for that purpose.

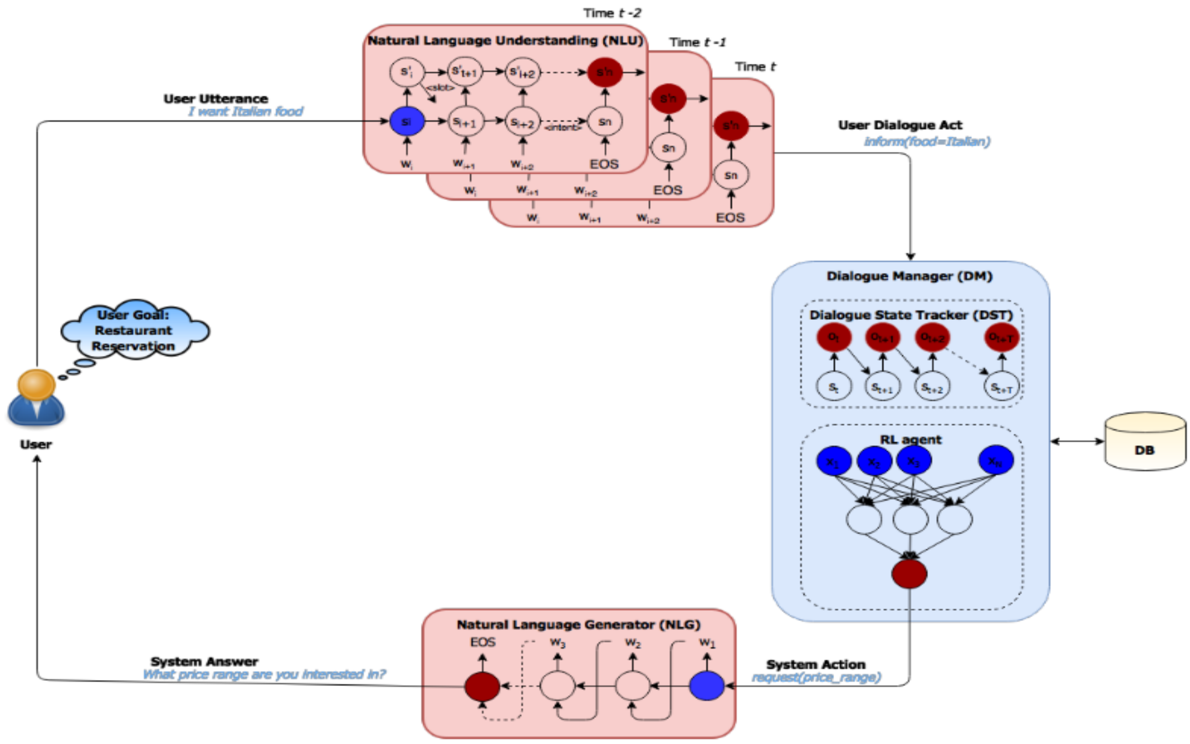

Figure 4.2 : RL bot structure [2]

4.2.3 Hybrid methods

This option is the combination of supervised and reinforcement learning based approaches, so that in order to use positive sides of both methods. In this method, supervised learning is used to restrict the model for avaliable data, while the reinforcement learning is used to optimize learnt dialogue policy [35].

In [36], the dialogue policy is trained in two phase. In first phase, the policy is trained on a fixed data in supervised learning method. In second phase, the trained policy is fined tuned by reinforcement method.

In another study [19], GOB is trained in two modes: offline and online. In offline mode, the system is trained in supervised method by using LSTM and rule based templates. In online mode, the system is trained by using reinforcement methods and the system learns autonomously [2].

4.3 Applied GOB System

In this thesis, the main goal is to design a infobot about a specific domain such as restaurant searching. Therefore, closed domain dialogue system architecture is embraced. This type of dialogue systems are also called as Goal Oriented Bot (GOB).

The user searches a specific information, the system provides specific information in correspond to user searching.

In this study, end to end GOB approach is applied instead of conventional GOB approach. In conventional approach, the dialogue systems are built as a pipeline which has modules for language understanding, state tracking, action selection and language generation. This pipeline structure comes up with dependency, complexity and error propagation. Each module has to wait until previous module finishes its job. Another issue is the training procedure. Each module requires specialized labelled data to be trained [19]. On the other hand, end to end methods trains recurrent neural networks directly on dialog texts. RNN infers a latent representation that removes the need for state labels [19].

In this thesis, DeepPavlov [37] architecture which is inspired by Hybrid Code Networks [19] is used as end to end goal oriented bot. Applied dialogue system architecture is shown in Figure 4.3 and the general pipeline is as follows:

The user utterance is featurized using pre-built embedding model such as fasttext. Slot-filling and intent classification processes are applied, so that NLU job is completed.

Next, the output of slot values are passed to DST. The features and states are concatenated as feature vector. This feature vector is passed to RNN. The RNN computes the hidden state which is retained for the next timestep and then it is passed to dense layer and softmax. Hence the distribution over action templates is obtained. The results are normalized and then highest probable action is chosen so that the job of DM is completed.

The selected action and the dialogue state are passed to NLG templates. If it is avaliable, the dialogue state is passed to a database as well and the database result is passed to NLG templates. Based on dialogue state, selected action and the database result the system response is generated by using NLG template. Thus, NLG job is also completed.

In scope of GOB studies in the thesis, various neural network architectures and hyperparameter tunings have been applied. Different word embeddings have been generated. Some GOB experiments have been conducted to get better GOB system.

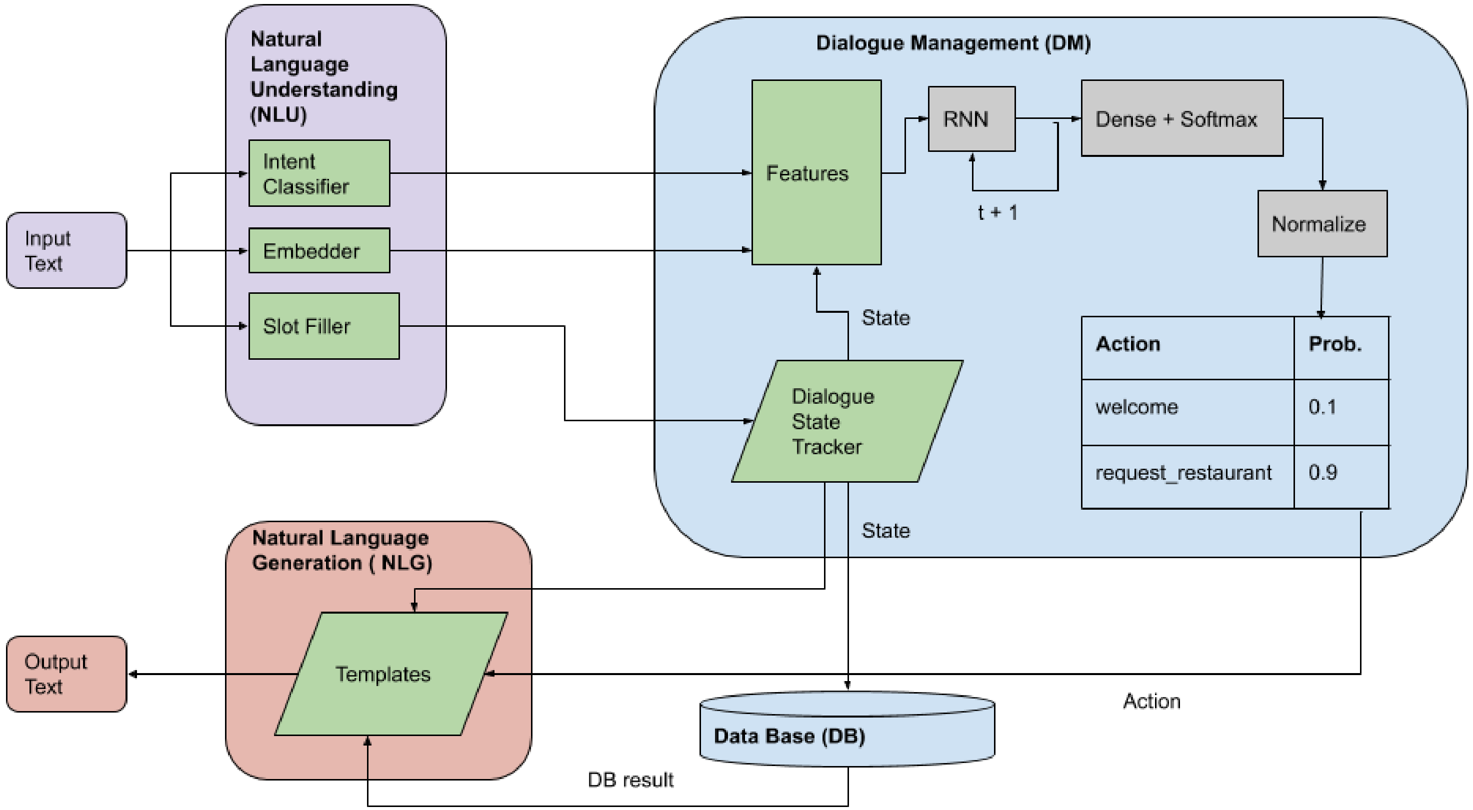

Figure 4.3 : Applied GOB structure

5. TEXT TO SPEECH (TTS)

Text to speech (TTS), also known as speech synthesis, is the system that converts text data into speech signal. This system provides talking ability to machines. This technology recently has gained major role in human life. It provides interactive environment for the user.

5.1 TTS Structure

A typical TTS system is consisted of four modules: text analysis, phonetic analysis, prosodic analysis, speech sytnthesis. The general pipeline of typical TTS system is shown in Figure 5.1.

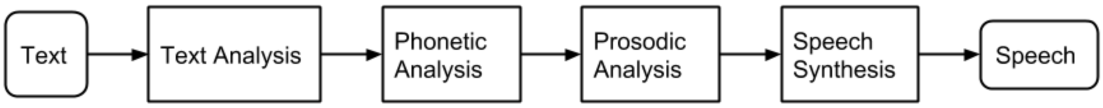

Figure 5.1 : Typical TTS structure

Text analysis module contains text normalization and linguistic analysis components. Text normalization means that numbers, symbols or abbreviations are replaced by their corresponding words. And punctutations are also removed. This process can be applied some sort of rule sets and regular expressipns. Linguistic analysis deals with syntactic and semantic analysis and aims to understand content of text since pronunciation of a word may depend on its meaning and on the context. Some machine learning methods can be applied to find the most proper meaning of the text.

Phonetic analysis module executes the grapheme-to-phoneme conversion. It is the process of converting a sequence of letters into a sequence of phones [15]. It means to generate a pronuncation for a word. For example, converting a letter string of *"CAKE"* into a phone string *"K EY K"*.

Prosodic analysis module attaches some prosodic features such as rythm of speech, strees patterns and intonation. Prosody has important effects on speech intelligibility and it gives some sort of speaker informations such as speaker's emotional state or social background. Besides that, prosody provides naturalness to synthetic speech. Some rules have been developed to attach pitch and duration features so that the system can gain prosody and more natural sounding.

Speech synthesis module is the final stage of TTS systems. Various methods can be applied for this stage. The important subjects of this stage are intelligibility and the continuoutity of the generated speech.

5.2 Speech Synthesis Methods

This section gives some information about existing speech synthesis methods: unit selection speech synthesis, statistical parametric speech synthesis, deep learning speech synthesis. Unit selection method is the most used speech synthesis system in commercial industries. Statistical parametric methods brings speech synthesis to new level. Deep learning methods present the state of art speech synthesis results.

5.2.1 Unit selection speech synthesis

This method is also called as concatenative method, because it is based on concatenating of speech segments. This system is built on real recorded speech databases. Using real speech segments gives better quality synthetic speeech, however this techniques comes up with memory management issue. The longer selected speech units causes fewer problem in concatenation process, however the memory requirements increase. Another issue is the coverage problem. The concatenated speech has a strong dependency on the chosen database, there is not much flexibility for new words. To handle this issue, diphone approach is developed. In diphone approach, the speech unit is selected from the middle of a phoneme to the middle of the next one as shown in Figure 5.2. In this figure, the phone boundaries are shown as blue dashed lines, the middle of the phones are shown as green lines. Selected unit is the area of two green lines. The reason of the middle point division is the fact that phone is more stable in the middle [3].

The general structure of the unit selection concatenative system is shown in Figure 5.3. In concatenation process, the units are selected is based on maximizing the prosodic match and minimizing the discontinuities that can appear during the concatenation of the speech units.

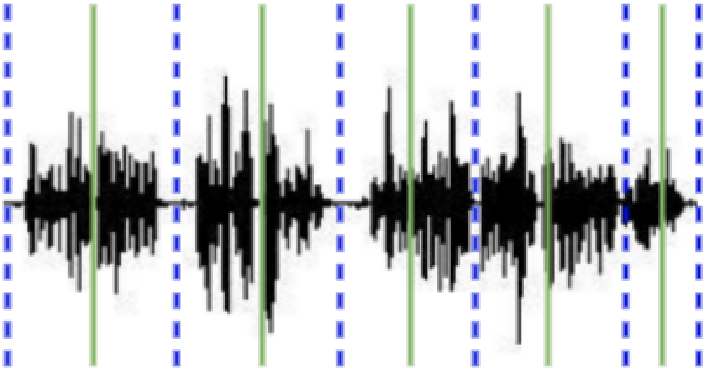

Figure 5.2 : Diphone TTS structure [3]

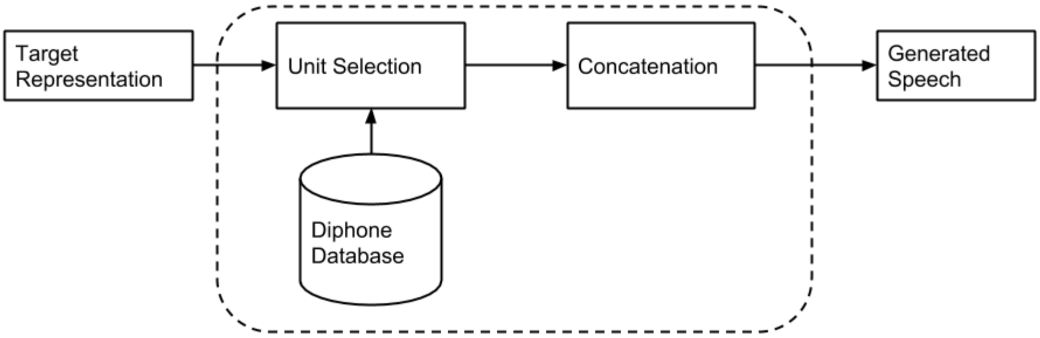

Figure 5.3 : Unit selection

To maximize the prosodic macth and minimize the discontinuities, speech databases are generated by single speaker through monotonical reading. Based on this database, generated speech is really intelligible because of real recorded speech, however the speech has robotic effect because there is no emphasis or emotion. Festival diphone synthesis [38] and MBROLA [39] can be shown as the example of the diphone speech synthesizers.

5.2.2 Statistical parametric speech synthesis (SPSS)

This TTS approach provides some kind of relation between text and acoustic frames by using generative acoustic models. In this approach, the speech is represented by some parameters and this TTS method tries to estimate by using statistical methods such as Hidden Markov Models (HMM). HMM based speech synthesis method is shown in Figure 5.4. This method consists of two phases: training and synthesis.

In training phase, acoustic features of speech and context features of corresponding text data are extracted from training database. Based on acoustic and context features, a parameter set of context-dependent HMMs (λ^*) is estimated by using Maximum Likelihood approach [3],

$$\lambda^* = argmax\ p(y|x,\lambda) \tag{5.1}$$

Where p(.) is a continuous probability density function (PDF), $y = \{y_1, y_2, ...y_T\}$ represents acoustic features sequence for each time frame whereas $x = \{x_1, x_2, ..x_N\}$ represents linguistic context features sequence for each phone. The acoustic feature vector includes static components and their first and second derivatives [40].

$$y = \{y_{st}, \Delta y_{st}, \Delta^2 y_{st}\} \tag{5.2}$$

The complete acoustic feature set at time t can then be considered as a linear transform over the static feature sequence $y_s = \{y_{s1}, y_{s2}, ..., y_{sT}\}$:

$$y = M_y y_s \tag{5.3}$$

M_y is determined by computing the first and second derivatives functions in equation (5.2).

There are large amount of context-dependent HMMs within linguistic context features in HMM-based speech synthesis so that this situation leads to data sparsity problem. For example, if there is some training example avaliable, the context dependent model can be overfits. However it causes the problem if the many valid combinations of the linguistic context features are absent from training database. To deal with data sparsity problem, a decision-tree-based clustering method is applied after initial training for clustering the state-output PDFs of the context-dependent HMMs [40]. A question set is designed for decision tree constriction by considering the characteristics of the processed language. The state alignment results wich use the HMMs are utilized to train the context-dependent state duration PDFs [41]. The state duration is modelled as Gaussian distribution in each state. A clustering method is applied to these duration models. Also, joint training of state output and state duration models are used [40].

The acoustic model $p(y|x,\lambda)$ used in HMM-based speech synthesis can be rewritten as

$$\begin{aligned} p(y|x,\lambda) &= \sum p(y,q|x,\lambda) \\ &= \sum P(q|x,\lambda)p(y|q,\lambda) \\ &= \sum P(q|x,\lambda)\prod_{t=1}^{T} p(y_t|q_t,\lambda) \end{aligned} \tag{5.4}$$

Where P(.) represents the probability mass function, $p(y_t|q_t,\lambda)$ represents the state output probability density function for q_t state. q_t can be modelled as single Gaussian distributed diagonal covariance matrix. q is the HMM state sequence of all q_t states. That's why, state duration probability $P(q|x,\lambda)$ is modelled within context-dependent Gaussian distribution for each state.

In synthesis phase, context features($\hat{x}$) of text are extracted by using front-end analysis. These features are passed through acoustic parameter generation process to determine the acoustic features which maximize the output probabilities of given text. After that, the vocoder syhnthesize the speech by using selected acoustic features.

$$y_s^* = argmax\ p(y|\hat{x},\lambda^*)|_{y=M_y y_s} \tag{5.5}$$

SPSS systems have a major advantage in flexibility compared to unit selection systems. Indeed speaker characteristic features such as duration of phonemes, expressiveness can be changed easily through some mathematical transformations. Speaker adaptation can be applied even the amount of train data is low. Furthermore, new speakers can be created through speaker interpolation technique which uses existing speaker models. Another advantage is the memory requirement. The storage needs are lower for SPSS compared to unit selection systems because after training, all needed component is the trained model. MaryTTS [42] can be shown as an example of SPSS system.

5.2.3 Deep learning speech synthesis

In recently, deep learning methods have also been applied to speech synthesis systems. The logic behind of the TTS is to map input string of text with output waveform. Deep learning models have already proved their efficiencies to learn proper features of input-output relations. That's why, deep learning can be a good alternative solution to SPSS methods for speech synthesis. Some promising results are obtained by using

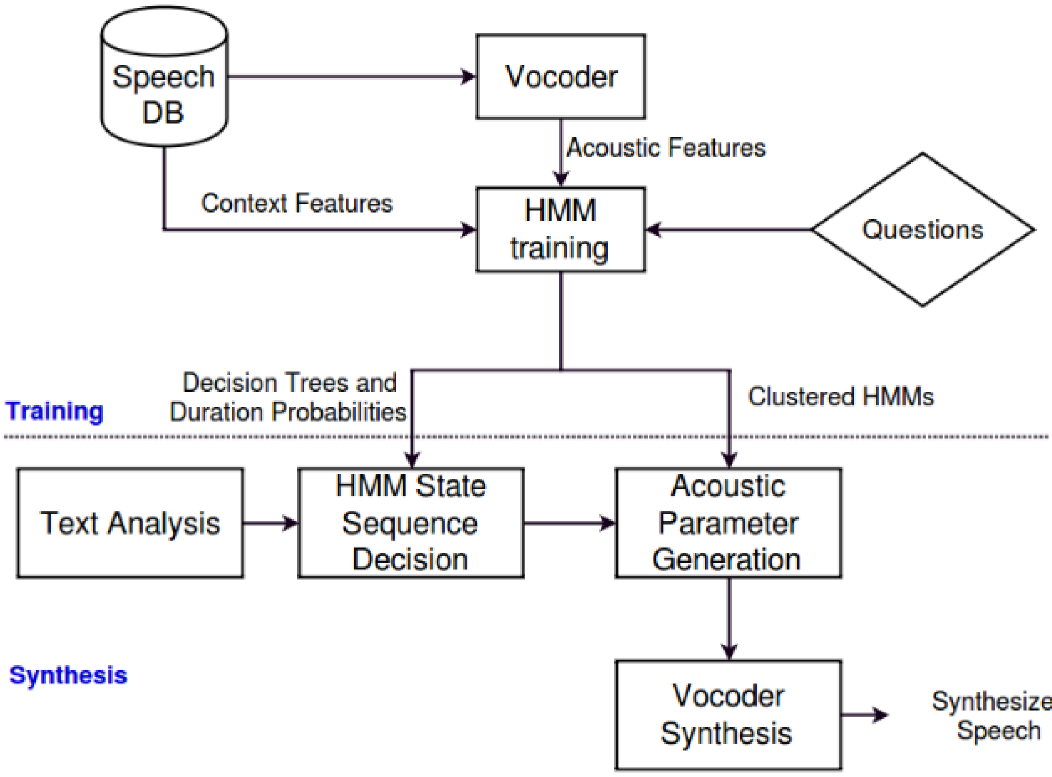

Figure 5.4 : Statistical parametric speech synthesis HMM-based system [3]

some deep learning approaches such as WaveNet, SampleRNN, Char2Wav, Deep Voice, Tacotron.

WaveNet [43] is a generative model of audio. WaveNet generates the audio sample by sample. Each sample is depends on previous samples generated in the audio. That's why, it has slow nature. It also requires conditioning on linguistic features from an existing TTS system. Wavenet is built on dilated casual convolutional layers.

SampleRNN [44] is also autoregressive generative model. However it is faster than Wavenet. SampleRNN has a hierarchical structure and consists of recurrent neural networks. In this approach, the input is segmented into chunks. From top to bottom, chunk size is decreased. Each upper level takes larger chunk of input , processes the chunk and passes it to lower level. The lower level processes in the same way and this situation continues until the bottom-most level in which a single sample is generated.

Char2Wav [45] is SampleRNN based speech synthesis with conditioning information on vocoder. This conditioning information is provided by using attention-based phone duration model and F0 prediction model.

Deep Voice is a deep learning based TTS approach which has three version:Deep Voice 1, 2 and 3. In Deep Voice 1 [46] and Deep Voice 2 [47], the conventional TTS pipeline is continued. Firstly grapheme to phone conversion is applied, next duration and frequency is predicted, finally waveform synthesis is occured. On the other hand, Deep Voice 3 [48] employs an attention based sequence-to-sequence model so that it has more compact architecture.

Tacotron [4] is end to end TTS system which is based on sequence to sequence architecture. Tacotron produces magnitude spectograms from a sequence of characters without knowing features and phone level alligment. Indeed, it does not require any existing TTS system to get the feature. Tacotron replaces the acoustic and linguistic feature generation with a neural network so that speech synthesis pipeline becomes simpler. The structure of Tacotron is consisted of four main components: CHBG module, encoder, decoder and post-processing module as shown in Figure 5.5.

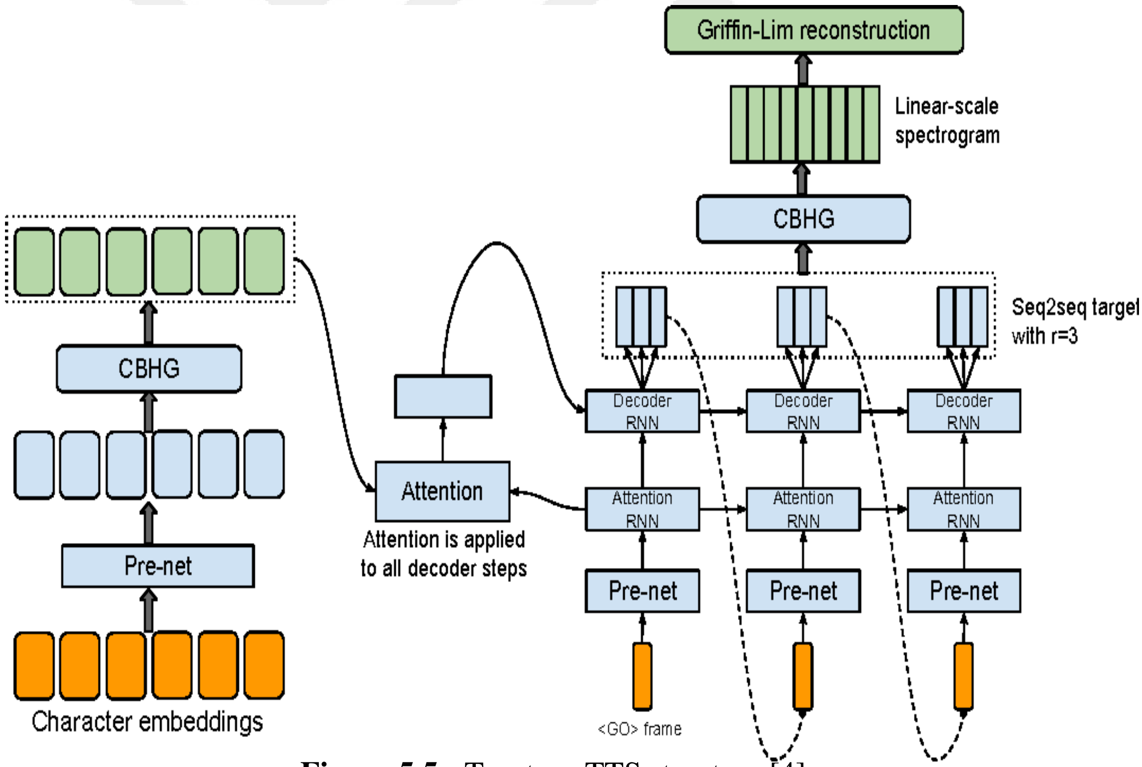

Figure 5.5 : Tacotron TTS structure [4]

CHBG module consists of 1D convolutional layer, highway network and bidirectional GRU as shown in Figure 5.6. In CHBG module, firstly the input sequence is convolved with a set of 1-D convolutional filters so that contextual information is modelled. The outputs of the convolution are stacked and max pooled along time. After that, some 1-D convolutions are applied to the processed sequence and the new outputs are added to the original input sequence by using residual connections. Batch normalization is applied for all convolutional layers. The convolution outputs are passed to multi-layer highway network in order to extract high level features. And finally, bidirectional RNN is applied to extract the sequential features from forward and backward direction [4].

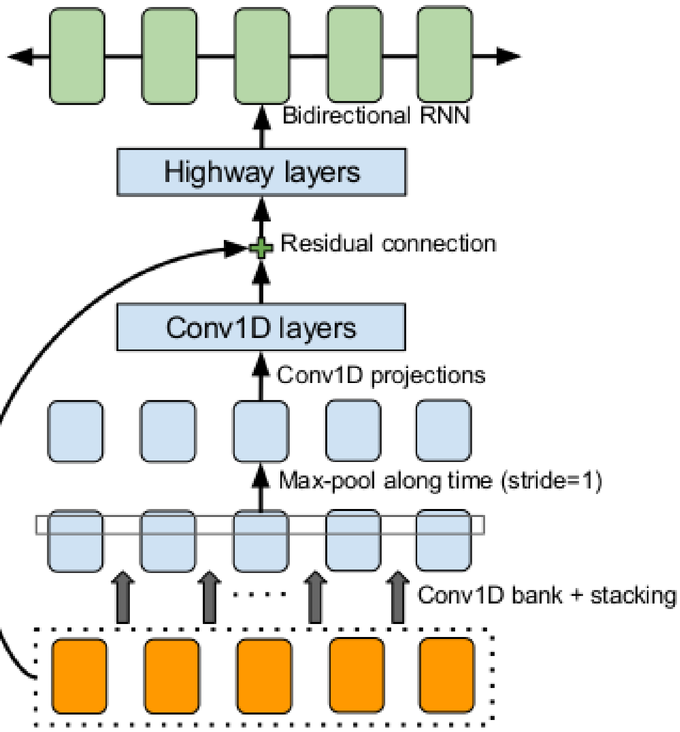

Figure 5.6 : CBGH structure [4]

Encoder takes embedding vector as input and gives encoded representation as output. Embedding vector is character sequence in which each character is represented as a one-hot vector. Non-linear transformations which are called as pre-net are applied to the each embedding [4]. The output of the pre-net is passed through CBHG module so that encoded representation is obtained for the attention module.

Decoder takes context vector generated by attention module as input and gives the spectogram frames. In this part, content based tanh attention decoder is used [4]. At each decoder time step, an attention query is produced by recurrent layers. In decoding process, at decoder step *t*, the last frame of the *r* predictions is fed as input to the decoder at step *t+1*. In Tacotron, multiple frames can be predicted at a time. The reduction factor *r* is the number of frames to predict at one step. If *r* increases, training and inference time decreases, aligment learning becomes easy. However, quality also decreases. That's why reduction factor should be tuned properly.

Post-processing module is fed with decoded r spectogram frames and it's output is target frame that can be synthesized into waveforms. The post-processing net learns to predict spectral magnitude sampled on a linear-frequency scale [4]. After that, predicted spectogram is synthesized to waveform by using Griffin-Lim algorithm [49].

5.3 Applied TTS System

In this thesis, end to end TTS approach is used instead of conventional TTS approach. Because conventional TTS systems requires multiple phases such as frontend text analysis, acoustic modelling and audio synthesis. Building these modules requires extensive domain expertise, extra time or major design changes. On the other hand, end to end TTS systems can handle these challenges. This system synthesizes the speech directly from characters. Within given text-audio pair dataset, the TTS model can be trained completely.

Tacotron based an end to end TTS system is applied in this study. However, it has important changes such as location sensitive attention and stop token prediction compared to Tacotron [4] architecture, so that training is faster and computationally very efficient.

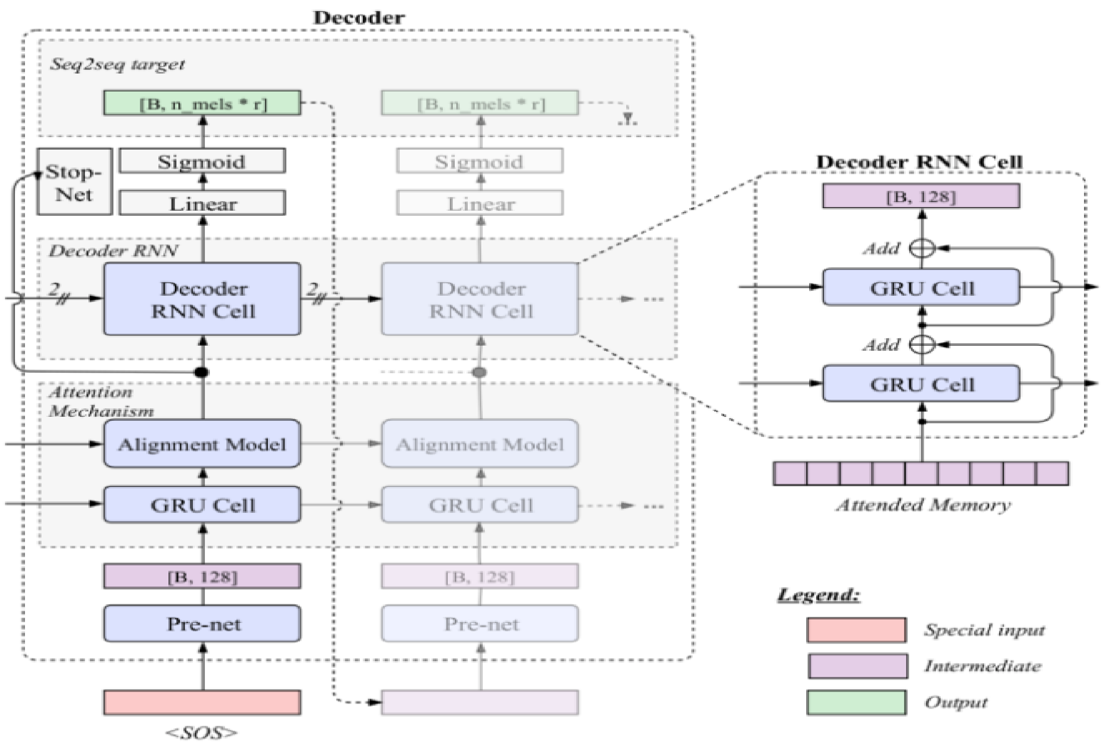

Figure 5.7 : Applied TTS decoder structure

In Figure 5.5, general Tacotron architecture is shown. In our applied modified Tacotron also has the similar arhitecture generally. Besides that, original Tacotron model does not propose a stop token to stop the decoding process, in modified Tacotron stop token prediction is provided within extra module as shown in Figure 5.7.

In scope of TTS studies in the thesis, some modifications have been applied Tacotron architecture to improve it's performance. Some TTS experiments have been conducted to improve TTS system.

6. EXPERIMENTAL STUDIES AND RESULTS

In this chapter, some experimental studies are conducted in scope of the thesis. Proposed spoken dialogue system (SDS) consists of three modules: automatic speech recognition (ASR), goal oriented bot (GOB) and text to speech (TTS). In following parts, experimental procedures and results are mentioned more detaily.

6.1 ASR Experiments

ASR is the first module of spoken dialogue system, so that it takes user speech utterance and converts it to text data and then that text data is passed to dialogue system. Since the output of ASR module is passed to dialogue system, the performance of ASR becomes important.

6.1.1 Datasets

In this study, ASR tests are conducted on three different English datasets: LibriSpeech [50], CommonVoice [51], Vystadial [52] datasets. All of these datasets are publicly avaliable. The amount of LibriSpeech dataset is about 1000 hours, the amount of CommonVoice dataset is about 200 hours and the amount of Vystadial dataset is about 41 hours.

6.1.2 Evaluation metrics

The performance of an ASR system is generally measured in terms of word error rate (WER) metric. It is ratio of total required editing word number between hypothesis and reference sequence to the word number of reference sequence. WER is calculated using equation (6.1).

$$WER = \frac{S+D+I}{T} \tag{6.1}$$

In equation (6.1), S represents the number of subtituted words, D represents the number of deleted words, I represents the number of inserted words and the T represents the number of words in reference sequence.

Another metric is character error rate (CER). It is similar to WER, but it is measured character level.

Lower WER value means better performance. Hence, the WER must be decreased to get better ASR result.

6.1.3 Experimental procedure and results

In this thesis, Deep Speech 2 based architecture is used for ASR system. The ASR training and testing experiments are conducted on WAV formatted audio files within 16 kHz sampling rate. Some of the thesis-default ASR parameters are presented in Table 6.1.

Table 6.1 : Default ASR parameters

Parameter	Value
frame length	20 ms
frame shift	10 ms
windowing	Hamming
sampling rate	16 kHz
RNN hidden layer number	7
layer node size	800
rnn type	gru
gpu usage	yes
learning rate	0.0003
optimizer	momentum
decoder	beam
α	0.8
β	1

In this thesis, beam decoding is applied in ASR decoding to get better results. To do that, language model is required. Language model is generated by using training transcriptions of datasets. 3-gram language model is embraced through this thesis.

Firstly, an ASR system is trained by using LibriSpeech dataset to compare the our model with original Deep Speech 2 model. In LibriSpeech experiment, the acoustic model is trained by using 960 hours training data, the languge model is generated by using training data transcriptions and the ASR test is conducted about 5 hours clean test data. Original model shows better performance compared to our trained model

Table 6.2 : ASR LibriSpeech clean test result

Model	Test Data	WER(%)	CER(%)
Our model	LibriSpeech test-clean	8.363	3.219
Original DS2 model [27]	LibriSpeech test-clean	5.33	-

as shown in Table 6.2, but still our implemented algorithm can be acceptable. This performance difference can be explained with used language model differences or some sort of optimization techniques. This comparison gives reliability information for implemented ASR algorithm.

One of the cricitical ASR problems is the accent. The accent of speaker can affects ASR performance. LibriSpeech is audio book dataset, so that it is relatively clean in accent perspective. In real world, there are different type of accents and this situation can affect speech recognition performance negatively.

In order to handle accent issue, another acoustic model is trained by using CommonVoice dataset. CommonVoice dataset includes speech data from different countries, so that it includes more accented data and more acoustic variability. Another language model is also trained by using CommonVoice train data transcriptions. The experiments are continiuned on CommonVoice dataset.

End to end ASR systems include two major module: acoustic model and language model. Acoustic model training requires much more time, data and effort compared to language model training. Therefore acoustic model training becomes more important. Indeed, an implicit language model is also trained beside of acoustic model in our system. The trained acoustic model can be used for ASR barely which is called greedy decoding, on the other hand language model can not be used barely. Since acoustic model training becomes more critical for ASR, next experiments focuses on acoustic model training.

Acoustic model can be affected from various elements such as model structure, train data type or amount etc. In following experiment, ASR performance is examined in acoustic model depth perspective. In this experiment, the acoustic model is trained by using about 200 hours CommonVoice training data, the languge model is generated by using training data transcriptions and the ASR test is conducted about 5 hours data. According to Table 6.3, deeper acoustic model shows beter performance on ASR.

Table 6.3 : Acoustic model depth effect on ASR

Test Data	Acoustic Model RNN layer	WER(%)	CER(%)
CommonVoice test	3 layered	8.942	5.340
CommonVoice test	5 layered	6.340	3.687
CommonVoice test	7 layered	5.144	2.995

Acoustic variability is an important factor for acoustic modelling. Training with different acoustic typed data can provide more general and more robust acoustic model. If extra data is avaliable, simplest solution is adding extra data. However if there is no extra data, some sort of augmentation techniques can be applied to increase the acoustic variability. To change the acoustic variability of the audio, speed or volume perturbation methods can be applied as data augmentation. Hence, the amount of data and the acoustic variability are increased. In next experiment, the speed of audio is perturbated in between 0.85 and 1.15 ranges and the volume of audio is perturbated between in between -6 dB and 8 dB as uniformly random distributed. Just acoustic variability is increased while the amount of total data stays same to prevent extra storage requirement. According to Table 6.4, augmented model has little bit performance loss. This can be caused from acoustic variability difference of train and test data. In augmented experiment, the training data is perturbated data, on the other hand test data is clean data. Hence, the performance of augmented model is lower for clean test data.

Table 6.4 : Augmentation effect on ASR

Test Data	Acoustic Model Type	WER(%)	CER(%)
CommonVoice test	No augmentation	5.144	2.995
CommonVoice test	Augmented	5.330	3.220

Domain similarity is another factor which affects ASR performance. Vystadial dataset is the most similar dataset for our case since it includes *"restaurant, food, name, price range"* key values similar to our use case. Because of domain similarity, this experiment gives a possible insight for our use case. In this experiment, the acoustic model is trained by using about 40 hours Vystadial training data, the languge model is generated by using training data transcriptions and the ASR test is conducted about 1.5 hours data. And the ASR test result is presented in Table 6.5.

Table 6.5 : ASR Vystadial test result

Test Data	WER(%)	CER(%)
Vystadial test	6.816	3.376

In previously, the importance of domain similarity and accent are mentioned. In this experiment, some kind of transfer learning is applied to combine the acoustic variability power of accented data and the power of similar domain data. To do that, firstly an acoustic model is trained by using CommonVoice dataset. After that, acoustic model training is continued by using Vystadial dataset. Hence, the base model is obtained from CommonVoice (CV) dataset and then final model is obtained from Vystadial dataset. The effect of transfer learning for acoustic model can be shown in Table 6.6. According to results, %17 WER improvement is obtained by using transfer learning.

Table 6.6 : Transfer learning effect on ASR

Test Data	Acoustic Model Type	WER(%)	CER(%)
Vystadial test	Vystadial model	6.816	3.376
Vystadial test	CV-Vystadial transfer model	5.639	2.761

And the final ASR experiment aims to see the effect of train data increment. In this experiment, firstly LibriSpeech (LS) data is added to CommonVoice (CV) data to train acoustic model. After that, Vystadial is also added to combined data to train acoustic model. The tests are conducted on CommonVoice test data.

Table 6.7 : Data increment effect on ASR

Test Data	Acoustic Model Type	WER(%)	CER(%)
CommonVoice test	CommonVoice model	6.340	3.687
CommonVoice test	LS+CV model	5.062	2.851
CommonVoice test	LS+CV+Vystadial model	4.969	2.827

Adding extra data affects the ASR performance positively as shown in Table 6.7. According to results, %21 WER improvement is obtained by using data increment.

6.2 GOB Experiments

GOB is the second module and the brain of spoken dialogue system, so that it takes the output of ASR module and generates the system response. It includes multiple modules such as natural language understanding, dialogue state tracking, dialogue policy and natural language generation. Spoken dialogue system is built upon GOB module since the dialogue is consist of interpretation of input and then returning proper action response.

6.2.1 Dataset

In this study, GOB tests are conducted on Dialogue State Tracking Challenge (DSTC)2 dataset [31]. The domain of dataset is restaurant searching.

6.2.2 Evaluation metrics

This module includes different submodules such as intent classification, named entity recognition, slotfilling. Therefore different type of evaluation metrics are used in this part. This evaluations metrics are accuracy, precision, recall and F1-measure.

Accuracy: The ratio of correctly predicted observation to the total observations.

Precision: The ratio of correctly predicted positive observations to the total predicted positive observations.

Recall: The ratio of correctly predicted positive observations to the all observations in actual class.

F1-measure: The harmonic average of precision and recall.

6.2.3 Experimental procedure and results

GOB module includes multi submodules. For example, natural language understanding job is completed by using slotfilling, named entity recognition (ner) and intent classification mechanisms. Thesis-default GOB parameters are shown in Table 6.8. This parameters provide the best results.

Firstly, NLU experiments are completed. NLU is important to interpret the user intent for dialogue system. To do that, named entity recognition (NER), intent detection, slot filling experiments are conducted on DSTC2 dataset.

Within NER processing, the *area, food, price range* preferences of the user are indentified. The NER experiment results are shown in Table 6.9. In NER test, 7797 tokens are processed with 1185 phrases. 1182 phrases are found and 1152 of them correct.

Table 6.8 : Thesis-default GOB parameters

Parameter	Value
slotfill threshold	0.8
ner embedding dimension	100
ner net type	cnn
ner learning rate	0.01
ner hidden output size	64
intent tokenizer	nltk
intent embedding	fastText
intent net type	cnn
intent dense size	100
intent learning rate	0.1
intent optimizer	adam
gob tokenizer	spacy
gob embedding	fastText
gob net type	lstm
gob hidden size	128
gob dense size	128
gob learning rate	0.0003
gob optimizer	adam
gob attention hidden size	32
gob attention depth	3

Table 6.9 : NER results

Data	Precision(%)	Recall(%)	F1-measure(%)
area	97.22	99.06	98.13
food	98.23	97.37	97.80
price range	98.87	100	99.43
overall	97.46	97.22	97.34

After that, slot filling and intent classification experiments are conducted. The *"#price-range, #cuisine, #location"* slots are filled according to conversation states. The user intents are classified as *request,inform, greeting, unknown,confirm* mainly. Intent classification experiments are evaluated for more specific intents such as *"food request, pricerange inform, area confirm, hello"*. The system is tested over 28 intents. The best slotfilling and intent classification results are shown in Table 6.10.

Table 6.10 : Slot filling and intent detection best results

Test	Accuracy(%)
slot filling	96.83
intent classification	95.68

Finally, GOB experiments are conducted and the best result is presented in Table 6.11. That result is obtained from 576 example.

Table 6.11 : GOB best result

Test	Accuracy(%)
gob	56.22

While the performance of natural language understanding (NLU) is very good based on *intent classification, NER and slotfilling* results, the performance of GOB is not as good as NLU. This situation can be explained with low performance of the dialogue policy (DP) submodule.

Although the performance of GOB does not seem quite satisfying, it is the best result for DSTC2 dataset as far as we know. The literature results [37] are presented in Table 6.12.

Table 6.12 : GOB result comparisons

Model	Accuracy(%)
Bordes and Weston (2016)	41.1
Perez and Liu (2016)	48.7
Eric and Manning (2017)	48.0
Williams et al. (2017)	55.6
Deeppavlov(*)	55.0
Our gob(*)	56.22

In Table 6.12, (*) models can not be compared with the others directly, because of dataset partition differences. While (*) systems are developed with train/dev/test data partition, the others are developed with train/test data partition. However, it still gives information about performances of the systems.

The comparison of Deeppavlov and our gob model has interesting point. Although our gob model is based on Deeppavlov architecture, our model shows better performance by using better word embedding.

6.3 TTS Experiments

TTS is the last module spoken dialogue system, so that it takes the output of GOB module and vocalizes the system response.

6.3.1 Dataset

In this thesis, TTS system is trained by using LJSpeech dataset [53]. This dataset is publicly avaliable and it consists of 13100 audio file of a single speaker. Total amount of data is about 24 hours. In this dataset, each audio file is a mono-channel, 16-bit, PCM, WAV formatted with a sample rate of 22050 Hz.

6.3.2 Evaluation metrics

In TTS systems, two major factors are important. These factors are naturalness and intelligibility. Naturalness provides user more comfortable communication way. Intelligibility is important for the user to understand the system more easily. A TTS system can be evaluated by using several metrics such as Mean Opinion Score (MOS), Perceptual Evaluation of Speech Quality (PESQ) [54].

MOS: The result of subjective test obtained by using human listeners. This evalaution needs man power so that it can be expensive and it is impractical for recursive evaluation.

PESQ: An objective method designed for speech quality assessment of telecominication network and speech codecs. It compares the reference signal and test signal and applies listening test, after that PESQ algorithm returns a score between -1 and 4.5 range. Higher score is better for PESQ evaluation.

Since PESQ is objective evaluation metric and more practical for retesting, it is preferred as TTS evaluation metric in this thesis.

6.3.3 Experimental procedure and results

In scope of this thesis, English TTS system is developed by using LJSpeech dataset according to parameters which are shown in Table 6.13.

PESQ score gives information about speech quality. There is some relation between speech intelligibility and quality, however it is not neat to map the PESQ score to the speech intelligibility directly. While the speech quality is high, the speech intelligibility can be low. For example, a noise supression system can decrease the speech intelligibility while it increases the speech quality.

Table 6.13 : Thesis-default TTS parameters

Parameter	Value
frame length	50 ms
frame shift	12.5 ms
number of mels	80
sampling rate	22050 Hz
number of FFT point	1024
r	5
griffin lim iteration	60
power	1.5
gpu usage	yes
learning rate	0.002
optimizer	adam

Table 6.14 : PESQ scores for different synthetic dataset

Test Data	Average PESQ Score
LibriSpeech syntetic test	1.334
CommonVoice synthetic test	1.635
Vystadial synthetic test	1.355

For TTS evaluation experiments, ASR datasets are used. Test datasets are synthesized by using trained TTS system. The PESQ scores are obtained according to original and synthetic speeches. The PESQ results are shown in Table 6.14. The PESQ results of synthetic speeches take place in average regions of the PESQ ranges (-1 and +4.5). Hence, the quality perfomance of the TTS system can be evaluated as the average.

As an intelligibility metric, WER can be used to evaluate the speech intelligibility of the TTS system. To do that, firstly ASR test data transciptions are vocalized by using TTS system. After that, these synthetic test datasets are transcribed by using ASR system. The WER results of the real and systhetic datasets are shown in Table 6.15. The ASR results of the synthetic datasets are worse than the real datasets as expected. Since the acoustic models are trained with real datasets. The acoustic difference between real and synthetic dataset causes this perfomance difference. The WER of synthetic datasets are are approximately three fold of real datasets's WER, but the results are still acceptable if the acoustic difference is considered.

The WER result comparison gives some insight about speech intelligibility. When the speech intelligibility can be increased, the WER result differences can be decreased.

Table 6.15 : WER and CER comparison of real and synthetic speech

Test Data	WER(%)	CER(%)
LibriSpeech test	8.363	3.219
LibriSpeech syntetic test	28.226	14.405
CommonVoice test	5.144	2.995
CommonVoice syntetic test	13.975	7.778
Vystadial test	6.572	3.232
Vystadial syntetic test	21.072	11.980

6.4 SDS Experiments

In this part, a real world case study is conducted. The combined system is tested. The working flow of the system is as follows:

- The system listens to the user until the long silence.
- After that speech utterance is transcribed by ASR module.
- The transcribed data is passed to GOB module.
- GOB module interprets the transcribed data.
- According to interpretation, the system queries the database if it is required or the system selects the proper action.
- These proper actions are converted to sentence structure by using templates
- The system response is passed to TTS module and it is vocalized by TTS.

SDS experiments are conducted based on improvisation data. In first experiment, the system accuracy of SDS is evaluated. The user speech utterance is transcribed by ASR module and then transcribed data is pased to GOB module and finally the proper response is vocalized by TTS module. The proper responses of the SDS are examined with respect to user utterances. The average accuracy of the SDS is %76.47 as shown in Table 6.16. In next experiment, just GOB module is tested on same dataset. The same dataset is passed to GOB module directly as text input, so that the performance of GOB module is evaluated without ASR error.

In SDS experiments, the GOB module and SDS system have better perfomance on improvisation data compared to DSTC2 dataset. This situation can be occured from

Table 6.16 : SDS result

Test	Accuracy(%)
SDS (ASR+GOB)	76.47
GOB	92.15

dataset differences between DSTC2 test dataset and improvisation data. Another probable reason may be the similarity of improvisation data and DSTC2 train dataset, so that the system can favour the improvisation data and it can show better perfomance.

When the SDS and GOB performances are examined, it can be seen easily that the ASR module has quite effect on SDS performance. The ASR results of SDS can be examined in Table 6.17.

Table 6.17 : ASR results in SDS

Test	WER(%)	Accuracy(%)
ASR	19.67	60.78

In Table 6.17, WER represents the classical ASR evaluation metric, on the other hand, the accuracy represents the ratio of perfectly transcribed utterances to total utterances. In ASR accuracy evaluation, even a small character error is evaluated as wrong transcription. By comparing the SDS and ASR accuracies, it can be claimed that SDS can tolerate the some of ASR errors and it can give correct responses.

These SDS experiments are empirical. However, they still give some insight about the performance of the system.

6.5 Discussion

According to previous experimental studies, the results can be summarized as follows:

- Original Deep Speech 2 system is better than our implemented system for ASR, this situation can be caused from language model differences or some kind of optimization. Yet, our implementation is still reliable.
- Accent is a critical factor for ASR, training with different accented data can improve the robustness of the system.
- Better acoustic model provides better ASR performance.

- To obtain better acoustic model for ASR, the depth of acoustic model or the amount of train data can be increased.

- Domain similarity increases the ASR performance.

- The performance of natural language understanding (NLU) is pretty good. Although the performance of GOB does not seem quite satisfying, it is the best result for DSTC2 dataset as far as we know.

- According to PESQ and WER results in TTS experiments, the quality of synthetic speech is average, on the other hand the intelligibility level of the synthetic speech is acceptable.

- In real use case experiments, the GOB and SDS show better perfomance on improvisation data compared to DSTC2 dataset. This can be caused from dataset differences between DSTC2 test dataset and improvisation data. Another possible reason may be the similarity of improvisation data and GOB model training dataset, so that the system can be little bit biased for the improvisation data and it can show better perfomance.

6.6 Further Experimentation

During this thesis, besides English language, some experiments are conducted for Turkish language as well. However, these experiments are handled under extra work title. In this context, previosuly mentioned ASR and TTS architectures are adapted to Turkish language. Hence, Turkish ASR and TTS systems are developed easily as aimed.

For Turkish ASR part, Turkish news dataset is used. This dataset is a subset of the Turkish data which is used in study [20]. This dataset has been collected at Bogazici University in scope of TUBITAK research project. This dataset consists of broadcast news and TV channel news data [20]. The data used in Turkish ASR experiment consists of 16 bit, PCM, 16 kHz and WAV formatted audio files.

The amount of ASR training data is 100 hours and the amount of ASR testing data is 5 hours. The acoustic model is trained based on thesis-default ASR parameters as shown in Table 6.1. For language model, 3-gram language model is generated from

ASR training corpus. The language model includes approximately 190 000 words and approximately 3 million 3-grams. The ASR test result is shown in Table 6.18.

Table 6.18 : Turkish ASR test result

Test Data	WER(%)	CER(%)
Turkish TV test	19.947	6.513

For Turkish TTS part, Turkish microphone data which is read by a professional speaker is used. The recorded microphone data is 16 bit, PCM, 16 kHz and WAV formatted audio file. The TTS training data is collected from single speaker, total amount of data is approximately 18 hours.

Turkish TTS system is developed by using this microphone dataset according to parameters except sampling rate parameter which are shown in Table 6.13. Turkish TTS system is developed with 16 kHz data while English TTS is developed with 22 kHz data.

Within these Turkish ASR and TTS experiments, a small contribution is provided to Turkish speech processing literature. These experiments also prove one of the aims of this thesis. Two main modules of the developed system could be adapted to new language easily.

7. CONCLUSION

The objective of this thesis is to design an infobot system for specific domain. Restaurant searching area is hired as study domain in scope of this thesis. However, it can be adapted to new domains. The system aims to provide easiness and time-saving to people. To statisfy these purposes, it is decided that the system should have speech interface. Since speech is the most natural communication way of people, speech interfaced system satisfies this purpose. Hence the user does not have to deal with learning new technology, request writing or menu clicking. Another aim is to adapt this system for different domains easily. To do that, it is decided that the system has modular and reconfigurable structure. Within this ability, the system can be extendable for new features or it can be adapted to new domains and new languages. Thus, the aim of this thesis evolves to design a modular spoken infobot system.

Spoken infobot system corresponds to closed domain spoken dialogue system (SDS) in speech and language technologies field. That's why,a brief information about SDS is explained firstly. The components of the SDS are introduced. Typical SDS and the proposed SDS are presented. The advantages of proposed SDS are mentioned.

Next, automatic speech recognition (ASR) which is the first module of the SDS is explained. Typical ASR structure and the existing ASR methods are presented. Applied ASR system is introduced. Deep Speech 2 based ASR architecture is embraced in this thesis. The working principle of the applied ASR system is illustrated.

After that, goal oriented bot (GOB) which is the middle module of the SDS is introduced. Typical dialogue systems and applied dialogue system structure are compared. Applied GOB system is introduced. DeepPavlov based GOB architecture is embraced in this thesis. It has a hybrid structure which includes RNNs, templates and databases. The working principle of the applied GOB system is explained.

Thereafter, text to speech (TTS) which is the last module of the SDS is explained. Typical TTS structure, existing TTS methods and the some examples are presented.

Applied TTS system is introduced. Tacotron based TTS architecture is embraced in this thesis. However some modifications are added to the original Tacotron architecture to increase the speed and performance. The working principle of the applied TTS system is explained.

Lastly, experimental studies done in the scope of this thesis are presented. Used datasets and hyperparameters are mentioned. Experiment results are discussed.

To conclude that, this system is a combination of some hot research topics such as automatic speech recognition, goal oriented bot and text to speech. The developed system is open source and reconfigurable unlike some commercially well-known products. It can be modified easily for new domain, feature or language. This system supports offline working. All the processes are run locally, so that it provides information security. In summary, a modular spoken infobot system which can be addressed to different type of purposes developed in scope of this thesis.

REFERENCES

[1] **He, X. and Deng, L.** (2008). Discriminative learning for speech recognition: theory and practice, *Synthesis Lectures on Speech and Audio Processing*, *4*(1), 1–112.

[2] **Ilievski, V.** (2018). Building Advanced Dialogue Managers for Goal-Oriented Dialogue Systems, *arXiv preprint arXiv:1806.00780*.

[3] **Pascual de la Puente, S.**, (2016), Deep learning applied to speech synthesis.

[4] **Wang, Y.**, **Skerry-Ryan, R.**, **Stanton, D.**, **Wu, Y.**, **Weiss, R.J.**, **Jaitly, N.**, **Yang, Z.**, **Xiao, Y.**, **Chen, Z.**, **Bengio, S.** *et al.* (2017). Tacotron: Towards end-to-end speech synthesis, *arXiv preprint arXiv:1703.10135*.

[5] **Serban, I.V.**, **Lowe, R.**, **Henderson, P.**, **Charlin, L. and Pineau, J.** (2018). A survey of available corpora for building data-driven dialogue systems: The journal version, *Dialogue & Discourse*, *9*(1), 1–49.

[6] **Raux, A. and Eskenazi, M.** (2004). Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges, *InSTIL/ICALL Symposium 2004*.

[7] **McCauley, L. and D'Mello, S.** (2006). MIKI: a speech enabled intelligent kiosk, *International Workshop on Intelligent Virtual Agents*, Springer, pp.132–144.

[8] **Young, S.**, **Gašić, M.**, **Thomson, B. and Williams, J.D.** (2013). Pomdp-based statistical spoken dialog systems: A review, *Proceedings of the IEEE*, *101*(5), 1160–1179.

[9] **Jurčíček, F.**, **Dušek, O.**, **Plátek, O. and Žilka, L.** (2014). Alex: A statistical dialogue systems framework, *International Conference on Text, Speech, and Dialogue*, Springer, pp.587–594.

[10] **Bordes, A.**, **Boureau, Y.L. and Weston, J.** (2016). Learning end-to-end goal-oriented dialog, *arXiv preprint arXiv:1605.07683*.

[11] **Meena, R.** (2016). Data-driven methods for spoken dialogue systems: Applications in language understanding, turn-taking, error detection, and knowledge acquisition, *Ph.D. thesis*, KTH Royal Institute of Technology.

[12] **Turunen, M.** (2004). A Spoken Dialogue Architecture and its Applications, *Ph.D. thesis*, PhD Dissertation, University of Tampere, Department of Computer Science A-2004-2.

[13] **Wetjen, S.**, (2013), Dialog act recognition using dependency features.

[14] **Young, S.** (2002). The statistical approach to the design of spoken dialogue systems.

[15] **Jurafsky, D. and Martin, J.H.** (2014). *Speech and language processing*, volume 3, Pearson London.

[16] **Li, X.**, **Chen, Y.N.**, **Li, L.**, **Gao, J. and Celikyilmaz, A.** (2017). End-to-end task-completion neural dialogue systems, *arXiv preprint arXiv:1703.01008*.

[17] **Wen, T.H.**, **Gasic, M.**, **Mrksic, N.**, **Rojas-Barahona, L.M.**, **Su, P.H.**, **Ultes, S.**, **Vandyke, D. and Young, S.** (2016). Conditional generation and snapshot learning in neural dialogue systems, *arXiv preprint arXiv:1606.03352*.

[18] **Wen, T.H.**, **Gasic, M.**, **Mrksic, N.**, **Su, P.H.**, **Vandyke, D. and Young, S.** (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems, *arXiv preprint arXiv:1508.01745*.

[19] **Williams, J.D.**, **Asadi, K. and Zweig, G.** (2017). Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning, *arXiv preprint arXiv:1702.03274*.

[20] **Arısoy, E.** (2009). Statistical and discriminative language modeling for Turkish large vocabulary continuous speech recognition, *Ph.D. thesis*, Citeseer.

[21] **Dempster, A.P.**, **Laird, N.M. and Rubin, D.B.** (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

[22] **Forney, G.D.** (1973). The viterbi algorithm, *Proceedings of the IEEE*, *61*(3).

[23] **Karpagavalli, S. and Chandra, E.** (2016). A review on automatic speech recognition architecture and approaches, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *9*(4), 393–404.

[24] **Passricha, V. and Aggarwal, R.K.**, (2018). Convolutional Neural Networks for Raw Speech Recognition, From Natural to Artificial Intelligence-Algorithms and Applications, IntechOpen.

[25] **Graves, A. and Jaitly, N.** (2014). Towards end-to-end speech recognition with recurrent neural networks, *International conference on machine learning*, pp.1764–1772.

[26] **Graves, A.**, **Fernández, S.**, **Gomez, F. and Schmidhuber, J.** (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, *Proceedings of the 23rd international conference on Machine learning*, ACM, pp.369–376.

[27] **Amodei, D.**, **Ananthanarayanan, S.**, **Anubhai, R.**, **Bai, J.**, **Battenberg, E.**, **Case, C.**, **Casper, J.**, **Catanzaro, B.**, **Cheng, Q.**, **Chen, G.** *et al.* (2016). Deep speech 2: End-to-end speech recognition in english and mandarin, *International Conference on Machine Learning*, pp.173–182.

[28] **Heafield, K.**, **Pouzyrevsky, I.**, **Clark, J.H. and Koehn, P.** (2013). Scalable modified Kneser-Ney language model estimation, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pp.690–696.

[29] **Vinyals, O. and Le, Q.** (2015). A neural conversational model, *arXiv preprint arXiv:1506.05869*.

[30] **Weston, J.**, **Bordes, A.**, **Chopra, S.**, **Rush, A.M.**, **van Merriënboer, B.**, **Joulin, A. and Mikolov, T.** (2015). Towards ai-complete question answering: A set of prerequisite toy tasks, *arXiv preprint arXiv:1502.05698*.

[31] **Henderson, M.**, **Thomson, B. and Williams, J.D.** (2014). The second dialog state tracking challenge, *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp.263–272.

[32] **Sutton, R.S.**, **Barto, A.G.** *et al.* (1998). *Introduction to reinforcement learning*, volume135, MIT press Cambridge.

[33] **Dhingra, B.**, **Li, L.**, **Li, X.**, **Gao, J.**, **Chen, Y.N.**, **Ahmed, F. and Deng, L.** (2016). Towards end-to-end reinforcement learning of dialogue agents for information access, *arXiv preprint arXiv:1609.00777*.

[34] **Li, X.**, **Lipton, Z.C.**, **Dhingra, B.**, **Li, L.**, **Gao, J. and Chen, Y.N.** (2016). A user simulator for task-completion dialogues, *arXiv preprint arXiv:1612.05688*.

[35] **Henderson, J.**, **Lemon, O. and Georgila, K.** (2008). Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets, *Computational Linguistics*, *34*(4), 487–511.

[36] **Su, P.H.**, **Gasic, M.**, **Mrksic, N.**, **Rojas-Barahona, L.**, **Ultes, S.**, **Vandyke, D.**, **Wen, T.H. and Young, S.** (2016). Continuously learning neural dialogue management, *arXiv preprint arXiv:1606.02689*.

[37] **Burtsev, M.**, **Seliverstov, A.**, **Airapetyan, R.**, **Arkhipov, M.**, **Baymurzina, D.**, **Bushkov, N.**, **Gureenkova, O.**, **Khakhulin, T.**, **Kuratov, Y.**, **Kuznetsov, D.** *et al.* (2018). DeepPavlov: Open-Source Library for Dialogue Systems, *Proceedings of ACL 2018, System Demonstrations*, pp.122–127.

[38] **Taylor, P.**, **Black, A.W. and Caley, R.** (1998). The architecture of the Festival speech synthesis system, *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

[39] **Dutoit, T.**, **Pagel, V.**, **Pierret, N.**, **Bataille, F. and Van der Vrecken, O.** (1996). The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes, *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 3, IEEE, pp.1393–1396.

[40] **Ling, Z.H.**, **Kang, S.Y.**, **Zen, H.**, **Senior, A.**, **Schuster, M.**, **Qian, X.J.**, **Meng, H.M. and Deng, L.** (2015). Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends, *IEEE Signal Processing Magazine*, *32*(3), 35–52.

[41] **Yoshimura, T.**, **Tokuda, K.**, **Masuko, T.**, **Kobayashi, T. and Kitamura, T.** (1999). Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis, *Sixth European Conference on Speech Communication and Technology*.

[42] **Schröder, M.**, **Charfuelan, M.**, **Pammi, S. and Steiner, I.** (2011). Open source voice creation toolkit for the MARY TTS Platform, *Twelfth annual conference of the international speech communication association*.

[43] **Van Den Oord, A.**, **Dieleman, S.**, **Zen, H.**, **Simonyan, K.**, **Vinyals, O.**, **Graves, A.**, **Kalchbrenner, N.**, **Senior, A.W. and Kavukcuoglu, K.** (2016). WaveNet: A generative model for raw audio., *SSW*, *125*.

[44] **Mehri, S.**, **Kumar, K.**, **Gulrajani, I.**, **Kumar, R.**, **Jain, S.**, **Sotelo, J.**, **Courville, A. and Bengio, Y.** (2016). SampleRNN: An unconditional end-to-end neural audio generation model, *arXiv preprint arXiv:1612.07837*.

[45] **Sotelo, J.**, **Mehri, S.**, **Kumar, K.**, **Santos, J.F.**, **Kastner, K.**, **Courville, A. and Bengio, Y.** (2017). Char2wav: End-to-end speech synthesis.

[46] **Arik, S.Ö.**, **Chrzanowski, M.**, **Coates, A.**, **Diamos, G.**, **Gibiansky, A.**, **Kang, Y.**, **Li, X.**, **Miller, J.**, **Ng, A.**, **Raiman, J.** *et al.* (2017). Deep voice: Real-time neural text-to-speech, *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, pp.195–204.

[47] **Gibiansky, A.**, **Arik, S.**, **Diamos, G.**, **Miller, J.**, **Peng, K.**, **Ping, W.**, **Raiman, J. and Zhou, Y.** (2017). Deep voice 2: Multi-speaker neural text-to-speech, *Advances in neural information processing systems*, pp.2962–2970.

[48] **Ping, W.**, **Peng, K.**, **Gibiansky, A.**, **Arik, S.O.**, **Kannan, A.**, **Narang, S.**, **Raiman, J. and Miller, J.** (2017). Deep voice 3: Scaling text-to-speech with convolutional sequence learning, *arXiv preprint arXiv:1710.07654*.

[49] **Griffin, D. and Lim, J.** (1984). Signal estimation from modified short-time Fourier transform, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*(2), 236–243.

[50] **Panayotov, V.**, **Chen, G.**, **Povey, D. and Khudanpur, S.** (2015). Librispeech: an ASR corpus based on public domain audio books, *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*.

[51] Mozilla common voice, `https://voice.mozilla.org/en`, accessed: 2019-04-30.

[52] **Korvas, M.**, **Plátek, O.**, **Dušek, O.**, **Žilka, L. and Jurčíček, F.**, (2014), Vystadial 2013–English data.

[53] **Ito, K.** *et al.*, (2017), The lj speech dataset.

[54] **Rix, A.W.**, **Beerends, J.G.**, **Hollier, M.P. and Hekstra, A.P.** (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs, *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 2, IEEE, pp.749–752.

CURRICULUM VITAE

Name Surname: Ramazan Gökay

Place and Date of Birth: Antalya, 20.04.1991

E-Mail: gokayramazan@gmail.com

EDUCATION:

- **B.Sc.:** 2015, Yeditepe University, Faculty of Engineering and Architecture, Mechanical Engineering Department (Double Major)
- **B.Sc.:** 2014, Yeditepe University, Faculty of Engineering and Architecture,Electrical and Electronics Engineering Department

PROFESSIONAL EXPERIENCE:

- 2016-... : Researcher, TÜBİTAK BİLGEM
- 2014-2016 : Research Assistant, Yeditepe University

PUBLICATIONS ON THE THESIS:

- **Gokay R.**, Yalcin H., 2019. Improving Low Resource Turkish Speech Recognition with Data Augmentation and TTS. *16th IEEE International Multi-Conference on Systems, Signals and Devices 2019 (SSD 2019)*, Istanbul, Turkey.

OTHER PUBLICATIONS:

- Beke A., Saroğlu M., Kadem G., Cebe O., **Gökay R.**, Sezer V., 2016. Vehicle Platoon Control: Consensus Based Approach. *Turkish National Meeting on Automatic Control, Otomatik Kontrol Ulusal Toplantısı (TOK 2016)*, Eskişehir, Türkiye.
- **Gokay R.** , Masazade E., Erol Barkana D., Aydin C., 2015. Emotional State and Cognitive Load Analysis using Features from BVP and SC Signals. *IEEE 2015 International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2015)*, San Diego,California, USA.
- Aypar Y., Palaska Y., **Gokay R.**, Masazade E.,Barkana D., Sarkar N., 2014. Clustering of Emotional States under Different Task Difficulty Levels for the Robot-Assisted Rehabilitation System-RehabRoby. *11th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2014)*, Vienna, Austria.

- Aypar Y., Palaska Y., **Gokay R.**, Masazade E., Barkana D., 2014. Fuzzy Clustering of Emotional States for the Robot-Assisted Rehabilitation SystemRehabRoby. *36th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2014)*, Chicago, Illinois, USA.