UOC
Universitat Oberta
de Catalunya

Implementation of a Spoken Language System

Jessica Pérez Guijarro

Grau d’Enginyeria Informàtica

Àrea: Intel·ligència artificial

David Isern Alarcón

Carles Ventura Royo

5 Juny 2018

Licence

FITXA DEL TREBALL FINAL

Títol del treball:	Implementation of a Spoken Language System
Nom de l'autor:	Jessica Pérez Guijarro
Nom del consultor/a:	David Isern Alarcón
Nom del PRA:	Carles Ventura Royo
Data de lliurament (mm/AAAA):	5 Juny 2018
Titulació:	Grau d'Enginyeria Informàtica
Àrea del Treball Final:	Artificial Intelligence
Idioma del treball:	English
Paraules clau:	Automatic Speech Recognition Spoken Language Understanding Recurrent Neural Network

“Aburrimiento visceral, hipocondría, angustia cósmica, el punto más lejano del Sol, en mi órbita, el puro hastío de vivir es mi amarga tónica.”

KASE.O - Basureta (Tiempos raros)

Resum del Treball (màxim 250 paraules): Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball

Este proyecto consiste en la implementación de un sistema de lenguaje que forma parte de los sistemas de diálogo como Siri. El sistema está constituido por dos bloques independientes: Automatic Speech Recognition, encargado de identificar aquello que está verbalizando el usuario y transformalo a texto, y el Spoken Language System, encargado de dotar comprensión al texto, identificando las partes significativas de dicho texto. Cada uno de los componentes se ha entrenado con técnicas y datasets distintos ya que, no comparten un objetivo común. En concreto, para el desarrollo del módulo ASR se ha trabajado con un subset previamente seleccionado del dataset VoxForge English, cuyos datos han sido entrenados mediante Hidden Markov Models. Por otro lado, para el desarrollo del módulo SLU se ha trabajado con Redes Neuronales Recurrentes y un variante del dataset ATIS previamente entrenado con el método Word Embedding. Pese a que la precisión obtenida en los cada uno de los componentes es más que aceptable, el funcionamiento de la integración de ambos resulta inestable.

Abstract (in English, 250 words or less):

This project consists of a implementation of a Spoken Language System that is part of the dialogue systems like Siri. The system is constituted by two independent blocks: Automatic Speech Recognition, in charge of identifying who is verbalizing the user and transforming a text, and the Spoken Language System, in charge of reading the text, identifying the significant parts of said text. Each of the components has been trained with different techniques and datasets since it does not share a common goal. In particular, for the development of the ASR module we have worked with a previously selected subset of the VoxForge English dataset, whose data has been trained using Hidden Markov Models for generate the Acoustic Model. On the other hand, for the development of the SLU module we have worked with Recurrent Neural Networks and a variant of the ATIS dataset previously trained with the Word Embedding method. Although the precision obtained in all the components is more than acceptable, the performance of the integration of both components results unstable.

Contents

List of Figures

1. Introduction

1.1. Context and justification of the thesis

Speech is an act of will that involves a certain degree of intelligence on the part of the being that executes the action. It is the most natural way of communication of the human being and still today a challenge for machines, due to the inherent complexity of the whole process.

In the 50s Alan Turing proposed the possibility of the existence of thinking machines with the ability to imitate human behavior, such as holding a conversation [16]. He developed the Imitation Game, also known as Test of Turing. In this test, a player with the role of interrogator would have to evaluate the conversation held with two other players, a human and a machine, to determine which one is the machine.

In the 60s, appeared ELIZA (Weizenbaum, 1966) [8] one of the first chatbot dialog system in the history of Natural Language Processing and the first to pass the Turing Test. ELIZA was designed to simulate the behavior of a Rogerian psychologist, a Person-Center psychotherapy where the therapist is free to adopt the position of not knowing anything about the real world and therefore, not having to interpret what the patient saying. Thus, ELIZA was able to engage discourse but not to understand the content of the conversation. Despite this, many users claimed that ELIZA was gifted with intelligence, that it understood what was being said and even that it had feelings.

Decades later, Intelligent Personal Assistants (IPA) [1] emerged. These systems are able to communicate with the user through voice and carry out some specific tasks like giving information about flights from one specific city to another. Some of the most known systems are Siri, Alexa or Cortana.

Due to the growing importance of these systems in today's society, I have found it interesting to analyze and implement a Spoken Language System (SLS). Specifically, this project focuses on two of the components of a Spoken Dialogue System (SDS) [6][8][10]: the Automatic Speech Recognition (ASR) module [13, 8, 10] and the Spoken Language Understanding (SLU) module [19, 6, 18, 11]. The ARS module is responsible of transform the speech signal into words (text), and the SLU module is in charge of interpreting what the user expects from the system. The typical structure of a SDS is shown in Figure 1:

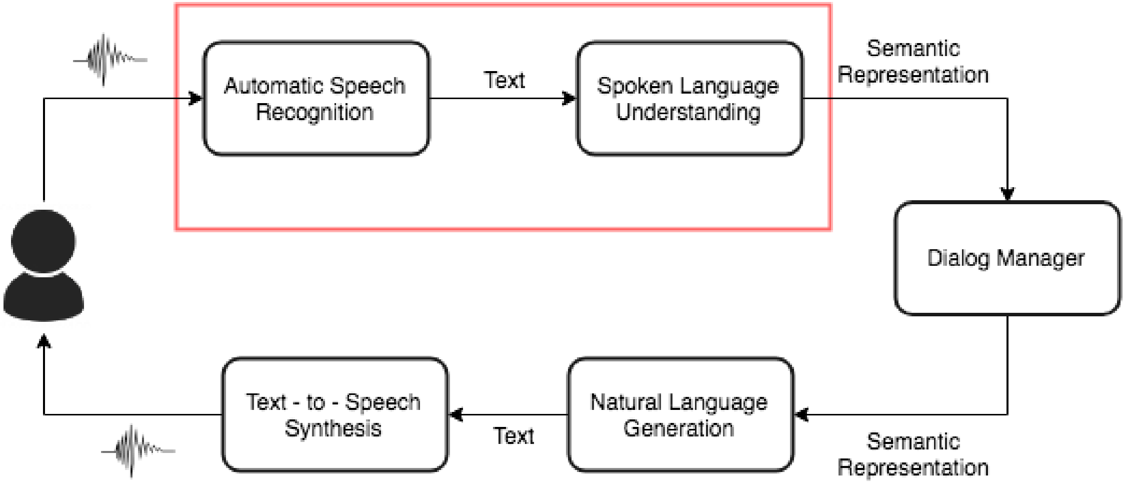

Automatic Speech Recognition
Text
Spoken Language Understanding
Semantic Representation
Dialog Manager
Semantic Representation
Natural Language Generation
Text
Text - to - Speech Synthesis

Figure 1: Structure of a Spoken Dialog System

1.2. Objectives of the Thesis

The purpose of this thesis is to develop an SLS that is able to recognize the user's voice command and transform it into a semantic representation of a specific knowledge domain; in this case, the chosen domain is Airline Traffic Information. The general objectives of the project are listed bellow:

1. Theoretical Natural Language Processing (NLP) introduction.

2. Research about methods and tools related to NLP.

The specific objectives of the project are the following:

- Design and implement the ASR and SLU modules of a Spoken Language System.

If there is enough time to spare the following objective is proposed:

- Build a web application to visualize the results of the Spoken Language System [18].

1.3. Followed strategy and methods

After the study of theory, current research and projects related to the chosen topic, a modular strategy has been chosen to carry out the project, where the development of the SLS system's components will be independently implemented.
In the following sections the methodology followed for the development of each module is detailed.

1.3.1. ASR module

The steps for the development of this module are:

- Search a speech transcriptions dataset
- Data preparation/preprocessing.
- Generation of the Acoustic Model.
- Generation of the Language/Grammar Model.
- Evaluation of the model.

The Acoustic Model is based on Hidden Markov Model with Gaussian Mixture Emissions (GMM - HMM). This technique has been chosen because it is one of the most cited in the consulted bibliography. In addition it performs and deliver better results than other techniques.
As for the Language Model (LM), it will be based on the n-grams probabilistic model. And finally, the performance evaluation of the generated model will be done using the Word Error Rate (WER) metric.

1.3.2. SLU module

The steps for the development of this module are:

- Search a dataset with a specific knowledge domain.
- Data preparation/preprocessing.
- Generation of the semantic model.

- Evaluation of the model.

The chosen method to implement the SLU is the statistical Semantic - frame, a data-driven approach whose main objective is to identify the semantic frames of the user utterance and extract the value of the slots associated with these frames.[1]

The generation of the semantic model is based on RNN with word embedding. It has been chosen to work with RNN because besides of being relatively a new methodology it achieves very good performance results in similar tasks to the proposal.

Finally, the performance evaluation of SLU module and its outputs (semantic representations) will be done using the Slot Precision or F1- Score metric

1.4. Thesis plan

1.4.1. Resources

In this section the resources that have been necessary for the development of this project are shown.

Python

Python is a general purpose interpreted programming language created in 1991 by Guido von Rossum. It is a stable language, widely used and with a multitude of libraries available.

HTK Toolkit

Htk Toolkits is a library designed to work with Hidden Markov Model (training and decoding) especially in the field of speech recognition. The toolkit has two different decoders, HViteand and HDecode, although none of them is designed for real - time decoding.

Continuous Speech Recognition Julius Engine

Julius Engine is a large vocabulary continuous speech recognition (LVCSR) decoder software based on the probabilistic method N-gram and context-dependent. In addition, it is capable of real-time decoding and can be easily integrated with HMMs and HTK Toolkit, which is why this tool has been chosen.

Keras

Keras is a library written in Python and designed to work faster with Neural Networks. It is also capable of running on other frameworks such as Tensorflow, Microsoft Cognitive Toolkit, Theano, or MXNet.

Datasets

In this thesis, the project will be done using two different datasets: VoxForge and ATIS.

VoxForge is an Open Source project that collects speech transcriptions recorded in different environments and in different languages. In this case, the chosen dataset is formed only with English recordings.

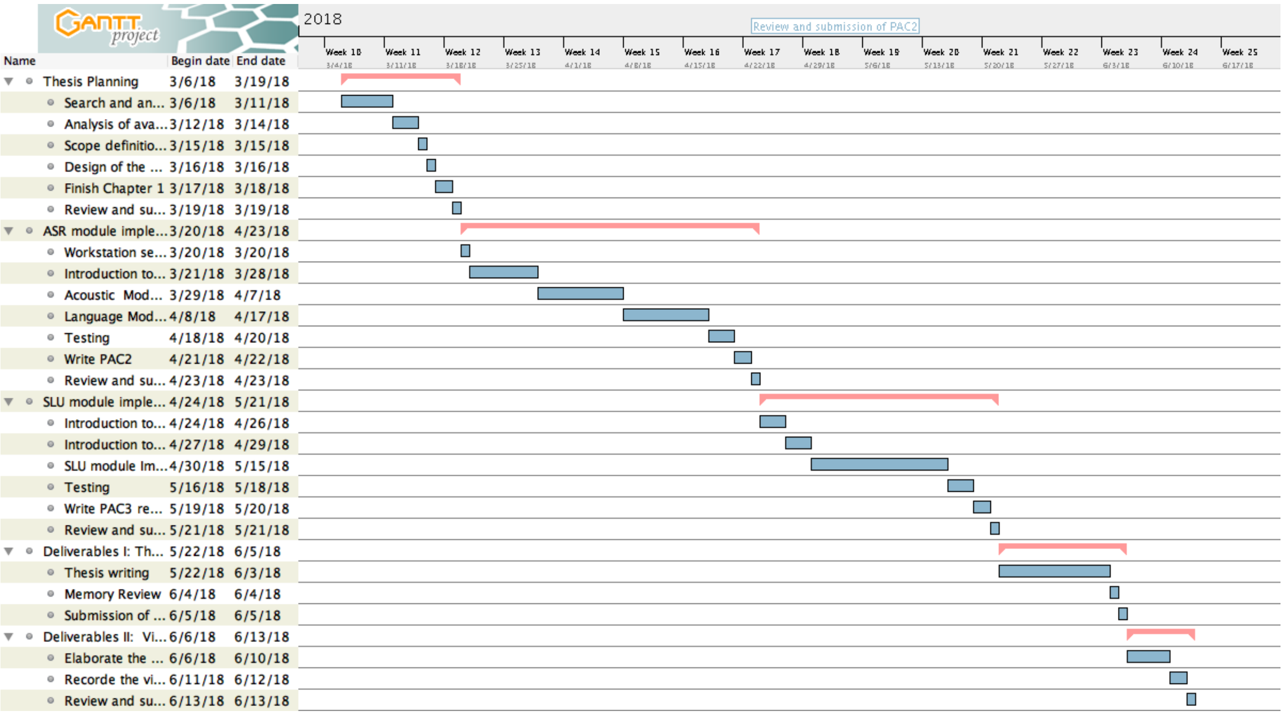
Air Travel Information System (ATIS) dataset has been chosen because it is the reference in multiple investigations on Natural Language Processing. In addition, the domain of the data is specific and limited to air traffic information, which reduces the complexity of the SLU module.

1.4.2. Time Planning

In this section is describe the temporal planning that will be followed throughout the project and the deliverables associated with each phase.

Phase	Task name	Duration	Start	End
Thesis Plan	**PAC 1**	**14**	**6/03/18**	**19/03/18**
	Search and analysis of information	6	6/03/18	11/03/18
	Analysis of available tools and technologies	3	12/03/18	14/03/18
	Scope definition and objectives	1	15/03/18	15/03/18
	Design of the project planning	1	16/03/18	16/03/18
	Finish Chapter 1	2	17/03/18	18/03/18
	Review of first version of the memory	1	19/03/18	19/03/18
ASR implementation	**PAC 2**	**35**	**20/03/18**	**23/04/18**
	Workstation setup	1	20/03/18	20/03/18
	Programming language and framework	8	21/03/18	28/03/18
	Acoustic Model Generation	10	29/03/18	7/04/18
	Language Model Generation	10	8/04/18	17/04/18
	Testing	3	18/04/18	20/04/18
	Write PAC2 report	2	21/04/18	22/04/18
	Review and submission of PAC2	1	23/04/18	23/04/18
SLU implementation	**PAC 3**	**28**	**24/04/18**	**21/05/18**
	Theory of Recurrent Neural Networks	3	24/04/18	26/04/18
	Introduction to Keras framework	3	27/04/18	29/04/18
	Semantic Model Generation	16	30/04/18	15/05/18
	Testing	3	16/05/18	18/05/18
	Write PAC3 report	2	19/05/18	20/05/18
	Review and submission of PAC3	1	21/05/18	21/05/18
Deliverable I Thesis	**PAC 4**	**15**	**22/05/18**	5/06/18
	Thesis writing	13	22/05/18	3/06/18
	Memory Review	1	4/06/18	4/06/18
	Submission of PAC4	1	5/06/18	5/06/18

Deliverable II	**PAC 5**	**8**	**6/06/18**	**13/06/18**
Virtual Presentation	Elaborate the virtual presentation	8	6/06/18	10/06/18
	Recorde the virtual presentation	2	11/06/18	12/06/18
	Review and submission of PAC5a	1	13/06/18	13/06/18



Name	Begin date	End date
▼ ● Thesis Planning	3/6/18	3/19/18
● Search and an...	3/6/18	3/11/18
● Analysis of ava...	3/12/18	3/14/18
● Scope definitio...	3/15/18	3/15/18
● Design of the ...	3/16/18	3/16/18
● Finish Chapter 1	3/17/18	3/18/18
● Review and su...	3/19/18	3/19/18
▼ ● ASR module imple...	3/20/18	4/23/18
● Workstation se...	3/20/18	3/20/18
● Introduction to...	3/21/18	3/28/18
● Acoustic Mod...	3/29/18	4/7/18
● Language Mod...	4/8/18	4/17/18
● Testing	4/18/18	4/20/18
● Write PAC2	4/21/18	4/22/18
● Review and su...	4/23/18	4/23/18
▼ ● SLU module imple...	4/24/18	5/21/18
● Introduction to...	4/24/18	4/26/18
● Introduction to...	4/27/18	4/29/18
● SLU module Im...	4/30/18	5/15/18
● Testing	5/16/18	5/18/18
● Write PAC3 re...	5/19/18	5/20/18
● Review and su...	5/21/18	5/21/18
▼ ● Deliverables I: Th...	5/22/18	6/5/18
● Thesis writing	5/22/18	6/3/18
● Memory Review	6/4/18	6/4/18
● Submission of ...	6/5/18	6/5/18
▼ ● Deliverables II: Vi...	6/6/18	6/13/18
● Elaborate the ...	6/6/18	6/10/18
● Recorde the vi...	6/11/18	6/12/18
● Review and su...	6/13/18	6/13/18

Figure 2: Gantt Time Planning

1.5. Brief summary of the obtained products

The purpose of this project is not to obtain a final product as such but rather to carry out a proof of concept.

1.6. Brief summary of other chapters in the memory

The structure of the document follows the order of the temporal planning presented in section 1.4.2.

- Chapter 2 *State of the Ar*t. Overview of the ASR and SLU components.

- Chapter 3 *Spoken Language System Design.* The proposed design to implement the SLS system is presented.

- Chapter 4 *Automatic Speech Recognition* This chapter covers several topics. On the one hand, it deepen into the methods used for the development of this module and on the other hand, it details the processes followed to achieve its implementation. Finally, the obtained results and performance will be discussed.

- Chapter 5 *Spoken Language Understanding*. This chapter will cover the same topics as the previous for the SLU module.

- Chapter 6 Conclusions and futures work This chapter presents the conclusions obtained with this project and discuss the possible improvements that can be developed in the future following the same line of work.

2. State-of-the-art

One of the most studied topics of the Natural Language Processing is the Spoken Language Systems, formed by the integration of two independent blocks and elaborated from very different and constantly evolving technologies.

The Spoken Language System are complex systems that don't depend on a single discipline but on the interrelation of several of them such as mathematics, linguistics or computer science. SLUs are responsible for very different tasks such as recognizing spoken utterances, understanding spoken commands or the ability to adapt to new speakers (intonation, accent ...).

As stated above, the SLSs are formed by two different blocks: ASR block and the SLU block. The function of the Automatic Speech Recognition component is to recognize and translate what a person says to text. For the development of this thesis, statistical models capable of solving problems related to speech recognition have been left aside to opt for a more modern approach where ASR systems use the information from language and acoustic model to select those most likely hypotheses.

Although current research focuses on acoustic models based on Deep Neural Network (DNN) or Recurrent Neural Network (RNN), this time it has been prefered a more traditional approach based on Hidden Markov Models (HMMs) since, they facilitate the task of modeling the acoustic relationships and the phonemes that define the language, all under a robust and highly efficient framework.

On the other hand, the function of the Spoken Language Understanding component is to correctly interpret what a person has said. The project follows a semantic-frame approach, characterized by working with a limited domain of knowledge and whose main structures are frames and slots. To build the language model that constitutes every SLU component, it has been opted to experiment with Recurrent Neural Networks a technique that has displaced Generative and Discriminative models over the last few years since it outstands performance in tasks related to natural language processing.[20]

So It's concluded that the main objective of this project is to reproduce using the most reliable and current techniques, a spoken language system constituted by the ASR and SLU components that allows the transformation of voice to text into real - time and understand the said text with the greatest precision possible.

3. Automatic Speech Recognition Module

The ASR module is responsible for recognizing the words a person has spoken and converting them to text. To carry out this process, the system needs a speech or audio signal recorded with the user's utterance input in order to process it and then generate the text corresponding to the pronunciation of the words.

The figure 3 presents an overview of the architecture design of the ASR module that will be detailed in the following points.

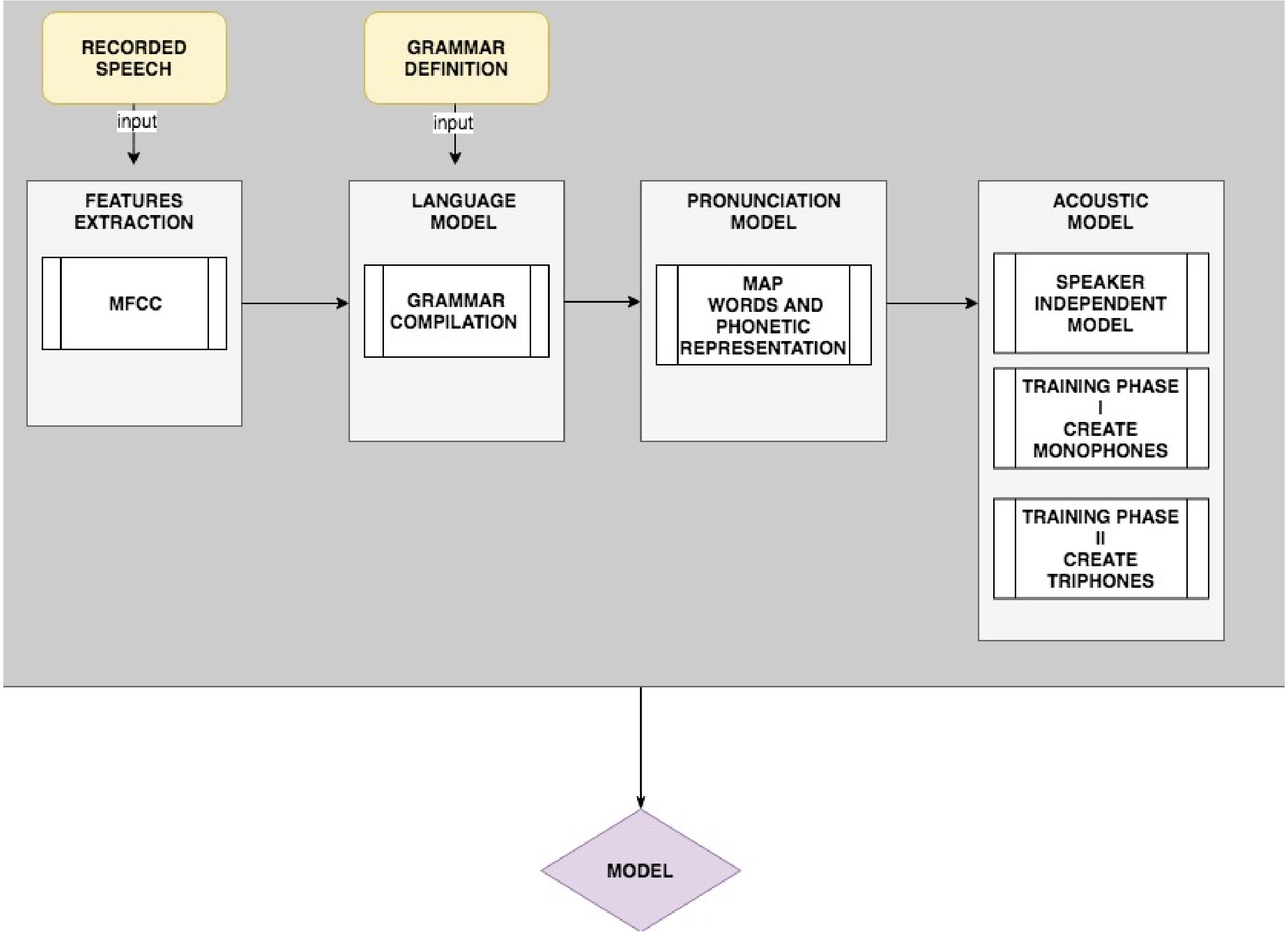


Figure 3: ASR Architecture

3.1. Data Preparation

3.1.1. VoxForge Dataset

The data used for the development of the ASR module comes from VoxForge project[11], specifically from VoxForge English Dataset. It consists of a total of 94351 samples of people of different gender and at least 4 different nationalities (New Zeland, Australia, USA, United Kingdom. For the development of the project 412 samples have been selected for training and 141 samples for testing.

The VoxForge project makes available audio files that users voluntarily record and submitted to the project. These audios are recorded in different conditions and environments and can be found in different languages.

All the data can be download for free and is compatible with any open source speech recognitions engines.

3.1.2. Data Recording

Because this module should work together with the SLU module, it has been considered convenient to add new audio files with information related to air traffic to the existing dataset.

For the recording of new audios, two subjects of different sex whose native language is Spanish have been chosen. In addition, the recordings have been carried out in a neutral environment without external noise and with specialized tools for the task.

It is important to note that audio files must be generated and saved in .wav format.

3.1.3. Transcriptions

The file prompts.txt is that file that contains the transcriptions of the audios included in the dataset. When adding new audio data it is necessary to add the corresponding transcriptions to the file along with the name of the folder where they are located. In Annex A there are the transcriptions of the audios that have been added.

3.2. Features extraction

Feature Extraction is the most important process of speech recognition process. As mentioned by HTK Book it is "parameterizise the raw speech waveforms into sequences of feature vectors" with the aim of deciphering the information and specific characteristics of the speaker contained in the speech signal of the submitted audio files. For this, it is necessary to convert audio .wav files to a new feature vector or MFCC format.

The MFCC technique for the extraction of features was introduced in 1980 Davis and Mermelstein[2] and still today is considered state-of-the-art for its effectiveness and robustness. It is based on the perception of the frequency range of the human ear and it consists of two types of filters: one for frequencies below 1000 Hz and another for frequencies higher than 1000 Hz.

It also integrates the advantages of the cepstrum analysis with a perceptual frequency scale method.

3.3. Rule - Based Language Modeling

The developed language model is based on the formal language theory of Chomsky; specifically, the grammar rules, which are those that allow to distinguish the valid syntactic structures from the domain of the system, are hand- written.

3.3.1. Grammar definition

The defined grammar that will determine which type of syntactic structures the ASR module will recognize is described in the following two files: vox.grammar and vox.voca.

vox.grammar

The file vox.grammar (Annex B) is responsible for specifying category rules and the structure of the allowed phrases.

Specifically, the grammar defined for this project consists of six word categories and is capable of recognizing 5 different syntactic structures. Each category defines a set of valid words that can be recognized by the acoustic model subsequently generated, along with their respective phonemes.

It is important to keep in mind that only those words that are included in the training dataset can be recognized by the defined grammar.

vox.voca

The file vox.voca (Annex C) registers under the name of a category such as NOUN or VERB each of the words that make up the domain of the model and its corresponding sequence of phoneme.

It is important to remember that to avoid compilation errors of the grammar the categories of both files must match.

3.4. Pronunciation Modeling

The pronunciation model or pronunciation dictionary is that model that provides us with a mapping between words and their phonetic representation. An example of an entry in the pronunciation dictionary it can be seen below.

AIRPORTS [AIRPORTS] eh r p ao r t s

For the development of the ASR module, an pronunciation dictionary has not been generated, but an existing one called VoxForgeDict has been worked on.

3.5. Acoustic Modeling

The acoustic model refers to that model that is built with the objective of representing the existing relationship between the audio and the phonemas of each of the words. To do this, a process is computed that assigns different statistical representations for the features vectors extracted from the waveforms of the audios with which we work. The most usual acoustic model and the one we use in this project is Hidden Markov Model (HMM).

Speaker Independent Model

The acoustic model developed is a speaker independent acoustic model or in other words, it is a model that is not designed to recognize only the speech of a person in particular but is able to recognize the speech of a person who has not even recorded any audio file of those used for the construction of the model.[7]

3.5.1. Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is an unsupervised classification method widely used in the field of speech and language processing.

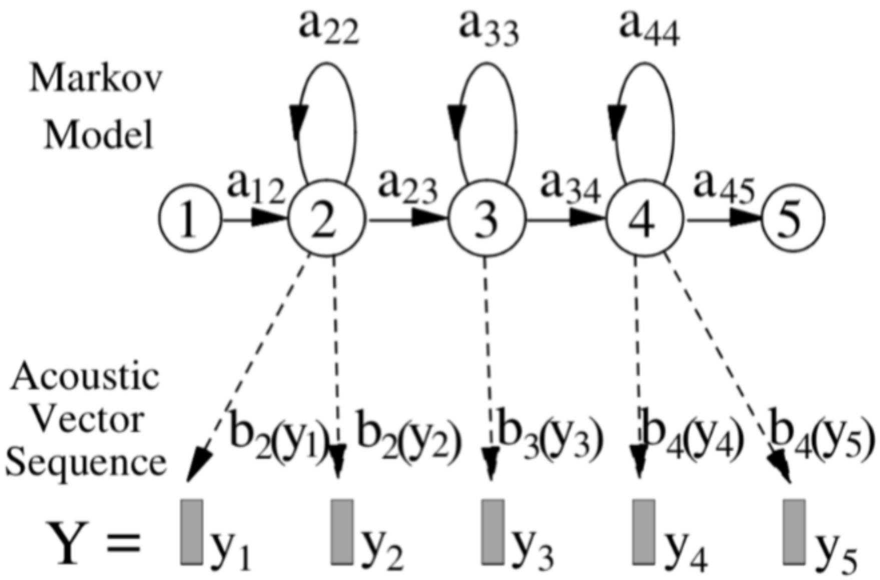


Figura 4: HMM for a phoneme sequence[4]

The HMM is a probabilistic sequence model whose main objective is to assign a class or label to each element of a sequence (words, letters ...), thus obtaining a sequence of observations mapped to a sequence of hidden states or labels. For this, the probability of distribution of the different labels or classes on each of the elements of the sequence is calculated, choosing the one that is considered the best candidate[4, 12, 17].

The most important components of HMM are the following:

- Set of N states. Unlike the output generated by the model, the states are not directly visible.
- Transition Probability Matrix that represents the probability of changing from one state to another.
- Set of O observations such as, for example, a set of phonemas.
- Set of observation likelihoods, also known as emission probabilities. Each observation likelihoods represents the probability that a state n generates an observation o.

3.5.2. Training phase

In the training phase, for each of the phonemes, an acoustic model based on HMM is generated that takes into account factors such as speaker variation, pronunciation variation or context dependent coarticulation variations.

Then, the steps that have been carried out to obtain the acoustic model by training the data with HMM are shown.[14]

Creating Monophones

First we use a prototype model to create a set of single gaussian monophone HMMs. Each of the HMM is defined by five states, where each of them is represented by single gaussian including the mean, variance and mixture weights of each of them. The first and last state define the model of silence [sil] that will be treated in later steps.

Then, a re-training of the previous set will be carried out during nine iterations to build a new HMMs whose values will have been updated to those that are more likely to be observed in the sequence.

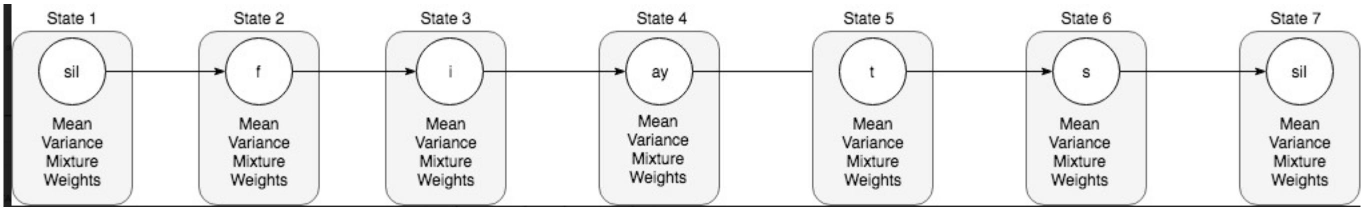


Figura 5: Monophone expansion of the word flights [4, 12, 17]

Using Monophones to create Triphones

Unlike the monophones where each word is represented phonema to phonema independently of the context, the triphone groups the phonemas of three in three depending on the context of each of them.

For the creation of the triphones or triphone transcriptions we start a process of reestimation of the monophone transcripcions that will result in a triphone HMMs.

Later, a process of tying based on decision trees will be carried out where all the triphones that are considered acoustically similar are tied with the aim of making the built model more robust.

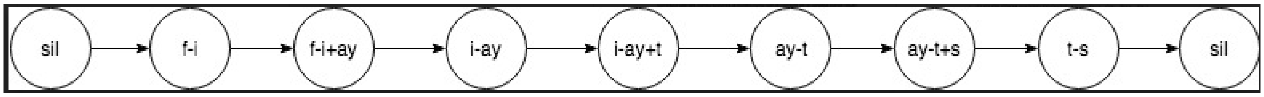


Figura 6: Triphone expansion of the word flights

3.6. Evaluation

3.6.1. Evaluation metrics

From the statistical indicator about recognition accuracy in relation to the sentences and words provided by the HResults tool included in the HTK ToolKit, it is possible to calculate the percentage of correct answers, accuracy percentage and the Word Error Rate (WER) metric. [4, 12, 17]

Where N is the total name of observations and D, I, S are the three errors that the system can make.

$$\text{(eq. 1)}\quad \%\ correct = \frac{N-D-S}{N} \cdot 100$$

$$\text{(eq. 2)}\quad \%\ accuracy = \frac{N-D-S-I}{N} \cdot 100$$

$$\text{(eq. 3)}\quad WER = \frac{D+I+S}{N}$$

Figura 7: Metrics

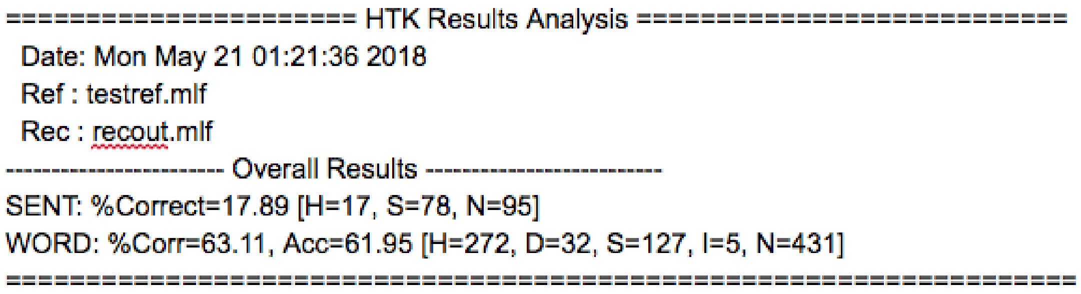
(D) Deletion: The system misses some word.

(S) Substitution: The system erroneously recognizes the word with a similar one.

(I) Insertion: The system adds extra words to the evaluated utterance.

3.6.2. Results

The results of the analysis of the ASR model built are shown below:



```
====================== HTK Results Analysis ===========================
  Date: Mon May 21 01:21:36 2018
  Ref : testref.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=17.89 [H=17, S=78, N=95]
WORD: %Corr=63.11, Acc=61.95 [H=272, D=32, S=127, I=5, N=431]
===================================================================
```

Figura 8: ASR module results

We observe that the model presents an accuracy of 61.95%. On the other hand, the probability of recognizing isolated words (63.11%) is higher than that of recognizing complete sentences (17.89% Correct). Finally, after calculating the WER metric (eq.3) it is considered that the global error of the system is 38.05%.

4. Spoken Language Understanding Module

The Spoken Language Understanding is the area of Language Processing that is responsible for providing understanding to the user utterances. In particular, the main task of the component is to detect and extract from the user utterance three determining factors to carry out satisfactorily the understanding[8]:

1. Domain Classification. This factor refers to the subject of what the user is talking about (books, food, transport ...)

2. Intent determination This factor refers to what the user's intention is. What do you intend to achieve? What is your objective?

3. Filling slot. This factor refers to the task carried out by the system to determine which slots and fillers need to be extracted from the user's intention to understand the user utterance.

Architecture

Below is an overview of the design of the SLU module architecture that will be developed in the following points.

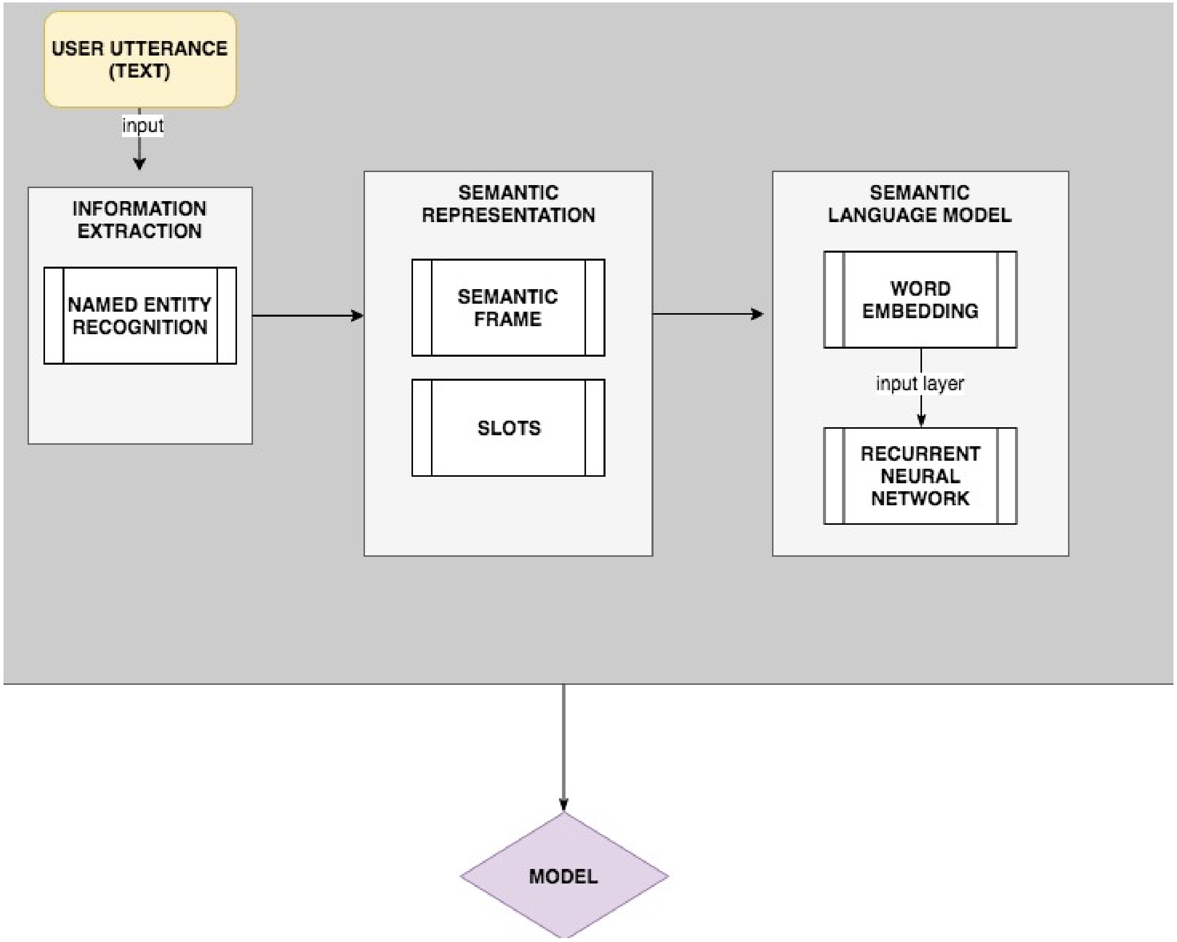
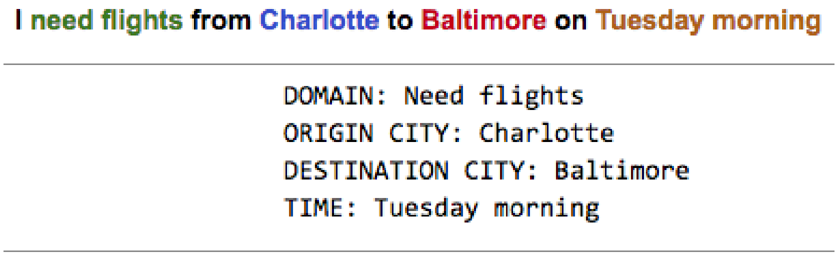


Figure 9: SLU Architecture

The SLU is responsible for providing understanding to the user's utterance. To carry out this purpose, we will train a Recurrent Neural Network with a previously tagged dataset following the IOB (Inside - Outside - Begging) representation format, which will define what type of tag will be assigned to each of the words that form the dataset. The Figure 10 shows which entities should the system recognize given a particular sentence.



I need flights from Charlotte to Baltimore on Tuesday morning

```
DOMAIN: Need flights
ORIGIN CITY: Charlotte
DESTINATION CITY: Baltimore
TIME: Tuesday morning
```

Figura 10: System classification perfomance

Challenges for Spoken Language Understanding

- Disfluencies: hesitations or false starts are some examples.
- Utterances that can not be modeled because they are out of the defined knowledge domain.
- Speech Recognition Errors due to the fact that the output generated by the ASR determines the modeling of the SLU.
- Problems with the syntactic structure of utterances.

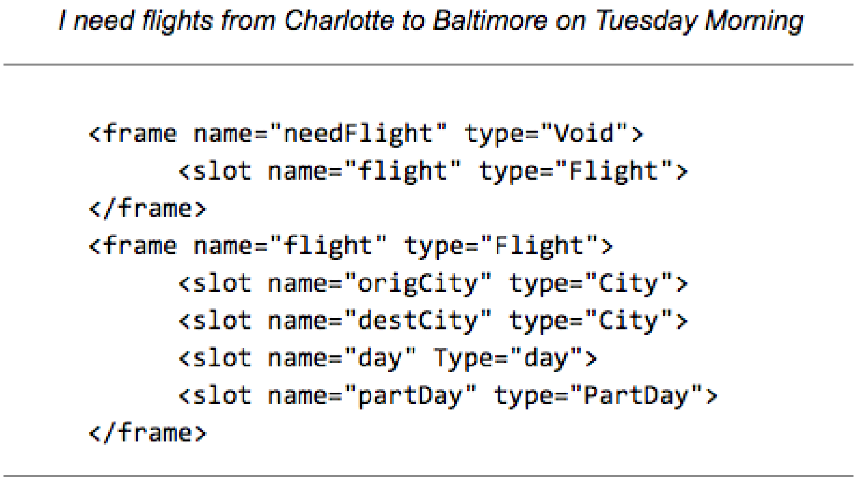
4.1. Information Extraction

The task of transforming the unstructured embedded information that composes any text into structured data is called information extraction (IE)[8]. More specifically, Information Extraction for Language Understanding aims to find values according to the slots previously defined and limited to a specific knowledge domain of given template. [3]
In the framework of the development of this project and with the intention of reducing the complexity of it, a pre - structured dataset has been chosen where the entities of the text have already an appropriate name or tag (flight, temporary event, place event ...) and therefore, no IE process will be done explicitly.

4.2. Semantic representation

The SLU component developed in this project bases the semantic representation of the components in a semantic frame approach whose main objective is to correctly select the semantic frames for a user utterance and then extract the key information for the slots. This task is called slot-filling, a special form of semantic parsing.

On the other hand, the semantic structure of the model is represented by semantic frames, where each frame contains different elements called slots. Each one of the slots must be filled according to the type of semantic information that is expected. It is important to bear in mind that this type of systems define a very limited and specific domain of knowledge since otherwise, the slot-filling task would be computationally more complex. For example, the figure 11 shows the semantic structure defined for the concrete example "I need flights from Charlotte to Baltimore on Tuesday morning", where the OrigCity slot (departure city) of the Flight frame can be filled with the semantic terminal City.



I need flights from Charlotte to Baltimore on Tuesday Morning

```
<frame name="needFlight" type="Void">
      <slot name="flight" type="Flight">
</frame>
<frame name="flight" type="Flight">
      <slot name="origCity" type="City">
      <slot name="destCity" type="City">
      <slot name="day" Type="day">
      <slot name="partDay" type="PartDay">
</frame>
```

Figure 11: Semantic - frame class scheme for the example “I need flights from Charlotte to Baltimore on Tuesday Morning

4.3. Semantic Language Model

Next, the steps that have been carried out for the develop the semantic language model whose main task is to predict words based on its lexical meaning, are explain in detail.

4.3.1. Atis Dataset

For the development of the SLU module, the ATIS dataset, the corpus most used in projects developed with SLU, will be used. This dataset collects information related to air traffic, such as information on flights, cities (destination and origin) or airports.

Specifically, ATIS collects a total of 4978/893 phrases and 56590/9198 words for train and test respectively, annotations of slot / concept, specifically 128 classes or slots including the label O (NULL), named entity, intent and the domain of knowledge of the data.[15]

4.3.2. Recurrent Neural Network (RNN)

Through the implementation of a Simple Recurrent Network (SRN) the semantic model has been built. It has been decided to feed the SRN with a previously trained input layer using the word embedding method that allows the clustering of words that are semantically similar. This process will not be explained in depth in this thesis since the chosen dataset had already been processed to obtain this end.

Model

The models based on Neural Networks relate a distributed representation of each word that make up the dictionary and calculate the joint probability of giving different sequences of words on said vectors.

The main input of the Neural Network is a - priori trained neural word embedding that consists of a one - hot representation of the next word of the input, where the index of the latter is represented by 1 and the rest is set to 0. (Figure 12). This action allows each of the words to be mapped to a data structure that provides a continuous-space representation.

$$\tilde{l}_i\,(\text{"}flight\text{"}) = \begin{bmatrix}0\\ \vdots\\ 1\\ \vdots\\ 0\end{bmatrix} \leftarrow \begin{array}{l}\text{The index of}\\ \text{word } \textit{flight} \text{ in the}\\ \text{vocabulary}\end{array}$$

Figure 12: Hot Encoding Representation[9]

The architecture of Elman, Jordan or Hybrid are some of the diverse architectures that exist to implement a Recurrent Neural Network. However, the chosen one for the development of this thesis is the Simple Recurrent Network (Elman 1990), one of the simplest RNN characterized by being formed with three layers and allowing cyclical connections of the network, unlike feed-forward back propagation network.

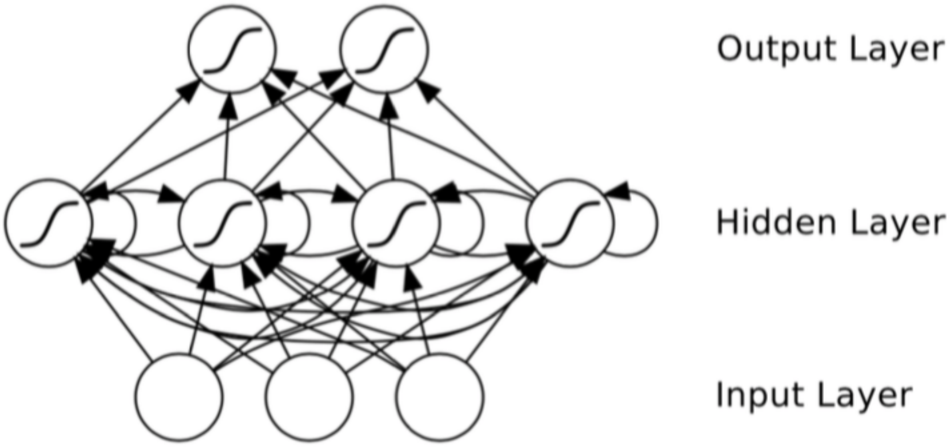


Figure 13: Recurrent Neural Network[5]

It's important to know that the RNNs determine how to respond to new data by analyzing the corresponding input, made up of the combination of the present input and the most recent past input. On the other hand, all models based on RNN include a hidden layer that stores information classified in the previous steps to produce a new hidden state and a new classification of information. This hidden layer, that simulates the "memory" of the RNN, will be connected to the output layer that calculates the estimated probability of each word in the dictionary using the softmax function, which represents a categorical distribution of the class labels with which we are working.

4.4. Evaluation

4.4.1. Evaluation metrics

The metrics used to evaluate the performance of the generated system are: loss and accuracy.

The Loss Function calculates the error that occurs between the difference of the real output and the prediction of the expected output.

$$\text{(eq.1)}\ Loss\ Function\ =\ p\ -\ \hat{p}$$

Figura 14: Loss Function Metric

The Accuracy metric calculates the number of correctly classified samples.

$$\text{(eq.2)}\ ACC\ =\ \frac{\Sigma\, true\ positive + \Sigma\, true\ negatives}{total\ samples}$$

Figura 15: Accuracy Metric

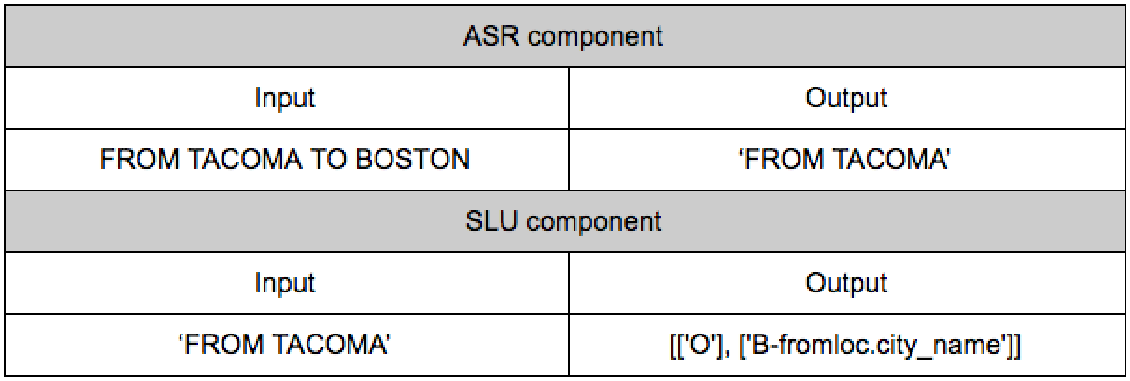
4.4.2. Results

After evaluating the SLU model developed, it is concluded that it is a robust and reliable model that allows to predict with 93.3 % accuracy and 33.3 % of loss the vast majority of the slots' value of a user utterance presented.

5. Experimental tests

Next, the tests carried out to evaluate the performance of the Spoken Language System or in other words, of the two integrated components and working jointly, are shown.

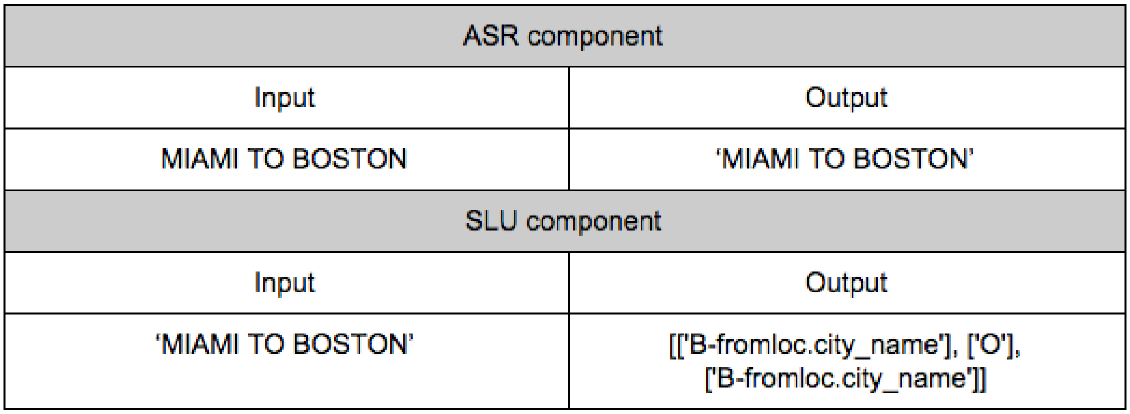
Test 1



ASR component	
Input	Output
FROM TACOMA TO BOSTON	‘FROM TACOMA’
SLU component	
Input	Output
‘FROM TACOMA’	[['O'], ['B-fromloc.city_name']]

We note that the ASR module recognizes partially the user input. This fact conditions the performance of the SLU module which, despite being able to generate a semantic representation of the user is biased and incomplete.

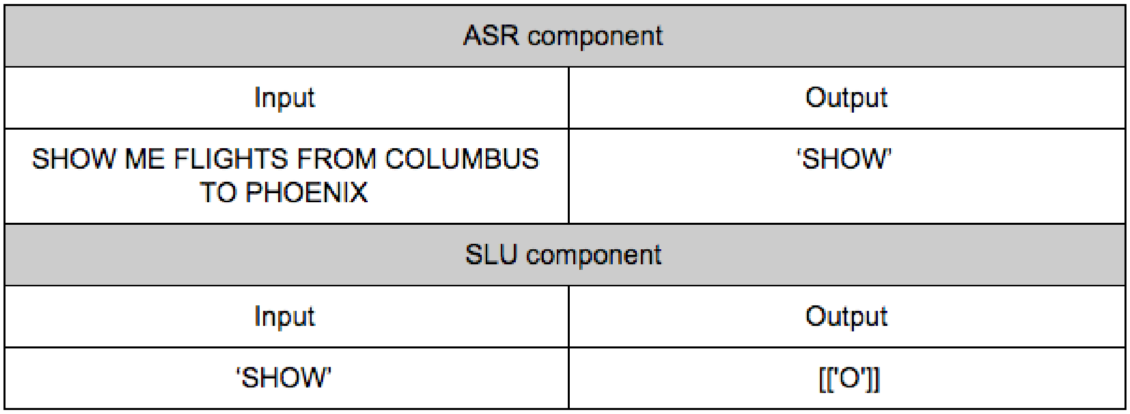
Test 2



ASR component	
Input	Output
MIAMI TO BOSTON	‘MIAMI TO BOSTON’
SLU component	
Input	Output
‘MIAMI TO BOSTON’	[['B-fromloc.city_name'], ['O'], ['B-fromloc.city_name']]

The ASR module recognizes the user’s complete utterance, which is transformed to a text and used as input to the SLU module. On this occasion, as the user utterance is faithful, we obtain a semantic representation according to the user’s original input.

Test 3



ASR component	
Input	Output
SHOW ME FLIGHTS FROM COLUMBUS TO PHOENIX	'SHOW'
SLU component	
Input	Output
'SHOW'	[['O']]

On this occasion, the ASR module has not been able to correctly identify any of the important parts of the sentence such as ’Columbus’ or ’Phoenix’. This fact can probably be caused by the length and complexity of the input. Consequently, the SLU module is not capable of generating a semantic representation that makes any sense.

6. Conclusions and Future works

In this project a Spoken Language System has been developed. It consists essentially on two independent components: ASR and SLU. Both components have been developed in parallel and using technologies and datasets with completely different characteristics. After experimenting and evaluating the system, a series of conclusions have been reached:

- The ASR module recognizes isolated words more efficiently than continuous speech since on one hand, it's easier to recognize the end points and on the other, the pronunciation of one word doesn't affect the next one. The experimental tests carried out support this conclusion since the experimental test with the worst results is the one that tries to recognize a complex and complete sentence.

- There are factors linked to the user that mess up with the performance of the ASR such as for example, the way of speaking, the intonation, the prosody or the pronunciation of the words.

- When we evaluate the performance of the two integrated blocks we see that the SLU module is not very efficient compared to its solo performance. This is because the input received by this module depends on the output generated by the ASR module, so if the latter is not able to correctly recognize what the user says, the SLU will not be able to generate a semantic representation faithful to the original input.

Regarding the future lines of work and research on the project, some of them are proposed:

- Introduce deep learning techniques such as Neural Networks for the development of the acoustic model of the ASR module with the intention of improving its performance and mitigate the effects of user speech (intonation, prosody ...).

- Expand the grammar of the language model of the ASR module, by a more extensive and complex grammar that allows to recognize more cities and other types of events such as, for example, temporary (days, parts of the day ...).

- Develop a dialogue system that incorporates the Spoken Language System created. Experiment with another type of Neural Networks such as LSTM for the development of the semantic model of the SLU module and compare which system has better performance.

7. Acronyms

IPA Intelligent Personal Assistants

NLP Natural Language Processing

SLS Spoken Language System

SDS Spoken Dialog System

ASR Automatic Speech Recognition

SLU Spoken Language Understanding

RNN Recurrent Neural Network

HMM Hidden Markov Model

8. Annexes

Annex A

carlos/mfc/en-227 AIRPORTS IN CHARLOTTE

carlos/mfc/en-228 AIRPORTS IN TACOMA

carlos/mfc/en-229 AIRPORTS IN ATLANTA

carlos/mfc/en-230 AIRPORTS IN PHOENIX

carlos/mfc/en-231 AIRPORTS IN BALTIMORE

carlos/mfc/en-232 AIRPORTS IN MONTREAL

carlos/mfc/en-233 AIRPORTS IN MEMPHIS

carlos/mfc/en-234 AIRPORTS IN BOSTON

carlos/mfc/en-235 AIRPORTS IN COLUMBUS

carlos/mfc/en-236 AIRPORTS IN MIAMI

carlos/mfc/en-237 AIRPORTS IN NASHVILLE

carlos/mfc/en-245 PHOENIX TO MIAMI

carlos/mfc/en-246 MIAMI TO BOSTON

carlos/mfc/en-247 MONTREAL TO COLUMBUS

carlos/mfc/en-248 TACOMA TO NASHVILLE

carlos/mfc/en-249 CHARLOTTE TO MIAMI

carlos/mfc/en-250 MEMPHIS TO BOSTON

carlos/mfc/en-251 ATLANTA TO BALTIMORE

carlos/mfc/en-252 MEMPHIS TO PHOENIX

carlos/mfc/en-253 FROM PHOENIX TO MIAMI

carlos/mfc/en-254 FROM MIAMI TO BOSTON

carlos/mfc/en-255 FROM MONTREAL TO COLUMBUS

carlos/mfc/en-256 FROM TACOMA TO NASHVILLE

carlos/mfc/en-257 FROM CHARLOTTE TO MIAMI

carlos/mfc/en-258 FROM MEMPHIS TO BOSTON carlos/mfc/en-259 FROM ATLANTA TO BALTIMORE

carlos/mfc/en-260 FROM MEMPHIS TO PHOENIX

jess/mfc/en-173 FLIGHTS FROM CHARLOTTE TO BALTIMORE

jess/mfc/en-174 FLIGHTS FROM ATLANTA TO BOSTON

jess/mfc/en-175 FLIGHTS FROM MIAMI TO MEMPHIS

jess/mfc/en-176 FLIGHTS FROM MONTREAL TO COLUMBUS

jess/mfc/en-177 FLIGHTS FROM NASHVILLE TO MONTREAL

jess/mfc/en-178 FLIGHTS FROM NASHVILLE TO PHOENIX

jess/mfc/en-179 FLIGHTS FROM TACOMA TO BALTIMORE

jess/mfc/en-180 FLIGHTS FROM MIAMI TO BOSTON

jess/mfc/en-181 FLIGHTS FROM PHOENIX TO ATLANTA

jess/mfc/en-182 FLIGHTS FROM CHARLOTTE TO TACOMA

jess/mfc/en-194 SHOW ME FLIGHTS FROM NASHVILLE TO BOSTON

jess/mfc/en-195 SHOW ME FLIGHTS FROM MIAMI TO TACOMA

jess/mfc/en-196 SHOW ME FLIGHTS FROM COLUMBUS TO PHOENIX

jess/mfc/en-197 SHOW ME FLIGHTS FROM ATLANTA TO MONTREAL

jess/mfc/en-198 SHOW ME FLIGHTS FROM BALTIMORE TO CHARLOTTE

jess/mfc/en-199 SHOW ME FLIGHTS FROM MEMPHIS TO BOSTON

jess/mfc/en-200 SHOW ME FLIGHTS FROM PHOENIX TO CHARLOTTE

jess/mfc/en-201 PHOENIX TO MIAMI

jess/mfc/en-202 MIAMI TO BOSTON

jess/mfc/en-203 MONTREAL TO COLUMBUS

Annex B

S : NS_B NOUN PREP CITY PREP CITY NS_E

S : NS_B NOUN PREP CITY NS_E

S : NS_B VERB AUX NOUN PREP CITY PREP CITY NS_E

S : NS_B CITY PREP CITY NS_E

S : NS_B PREP CITY PREP CITY NS_E

Annex C

```
% NS_B
<s> sil
% NS_E
</s> sil
% AUX
ME m iy
% VERB
SHOW sh ow
WANT w aa n t ah
% NOUN
FLIGHTS f l ay t s
AIRPORTS eh r p ao r t s
%PREP
TO t uw
FROM f r ah m
IN ih n
% CITY
BALTIMORE b ao l t ah m ao r
BOSTON b aa s t ah n
MEMPHIS m eh m f ih s
ATLANTA ae t l ae n t ah
CHARLOTTE sh aa r l ah t
MIAMI m ay ae m iy
MONTREAL m ah n t r iy ao l
PHOENIX f iy n ih k s
TACOMA t ah k ow m aa
COLUMBUS k ah l ah m b ah s
```

NASHVILLE n ae sh v ih l

Annex D

Those are the main instructions to generate the Acoustic Model:

```
#!Dir creation mkdir -p $HOME/voxforge/asr/bin mkdir -p $HOME/voxforge/asr/lexicon
ASR_PATH=$HOME/voxforge/asr
########!Step1 -> Download the data from www.voxforge.org
########!Step2 -> Complile the create grammar for the speech recognizer
#!Compiling grammar file
julia ../bin/mkdfa.jl grammarFile
#######!Step3 -> Create the Pronunciation Dictionary
julia ../bin/prompts2wlist.jl prompts.txt wlist
########!Step4 -> Create the Transcription files
julia ../bin/prompts2mlf.jl prompts.txt words.mlf
########!Step5 -> Pronuntiation Dictionary
HDMan -A -D -T 1 -m -w wlist -n monophones1 -i -l dlog dict ../lexicon/VoxForgeDict.txt
########!Step6 -> Training Monophones
HCompV -A -D -T 1 -C config -f 0.01 -m -S train.scp -M hmm0 proto
##Repeat the process
HERest -A -D -T 1 -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 monophones0
HERest -A -D -T 1 -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2 monophones0
HERest -A -D -T 1 -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm2/macros -H hmm2/hmmdefs -M hmm3 monophones0
HHEd -A -D -T 1 -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed monophones1 HERest -A -D -T 1 -C config -I phones1.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm5/macros -H hmm5/hmmdefs -M hmm6 monophones1
HERest -A -D -T 1 -C config -I phones1.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm6/macros -H hmm6/hmmdefs -M hmm7 monophones1
```

```
HVite -A -D -T 1 -l ’*’ -o SWT -b SENT-END -C config -H hmm7/macros -H hmm7/hmmdefs -i
aligned.mlf -m -t 250.0 150.0 1000.0 -y lab -a -I words.mlf -S train.scp dict monophones1> HVite_log
HERest -A -D -T 1 -C config -I aligned.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm7/macros -H
hmm7/hmmdefs -M hmm8 monophones1
########!Step7 -> Training Triphones
HLEd -A -D -T 1 -n triphones1 -l ’*’ -i wintri.mlf mktri.led aligned.mlf
julia ../bin/mktrihed.jl monophones1 triphones1 mktri.hed
HHEd -A -D -T 1 -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed monophones1
HERest -A -D -T 1 -C config -I wintri.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm10/macros -H
hmm10/hmmdefs -M hmm11 triphones1
HERest -A -D -T 1 -C config -I wintri.mlf -t 250.0 150.0 3000.0 -s stats -S train.scp -H hmm11/macros
-H hmm11/hmmdefs -M hmm12 triphones1
HDMan -A -D -T 1 -b sp -n fulllist0 -g maketriphones.ded -l flog dict-tri ../lexicon/VoxForgeDict.txt
julia ../bin/fixfulllist.jl fulllist0 monophones0 fulllist
cat tree1.hed > tree.hed
julia ../bin/mkclscript.jl monophones0 tree.hed
HHEd -A -D -T 1 -H hmm12/macros -H hmm12/hmmdefs -M hmm13 tree.hed triphones1
```

References

[1] Ali Orkan Bayer. *Semantic Language Models with Deep Neural Networks*. PhD thesis, University of Trento, Italy, 2015.

[2] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, Aug 1980. ISSN 0096-3518. doi: 10.1109/TASSP.1980.1163420.

[3] Jihyun Eun, Changki Lee, and Gary Geunbae Lee. An information extraction approach for spoken language understanding. In *in Proceedings of the 8 th International Conference on Spoken Language Processing*, pages 2145–2148, 2004.

[4] Mark Gales and Steve Young. The application of hidden markov models in speech recognition. *Found. Trends Signal Process.*, 1(3):195–304, January 2007. ISSN 1932-8346. doi: 10.1561/2000000004. URL http://dx.doi.org/10.1561/2000000004.

[5] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012. ISBN 978-3-642-24796-5. doi: 10.1007/978-3-642-24797-2. URL https://doi.org/10.1007/978-3-642-24797-2.

[6] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. ISBN 0130226165.

[7] J. Ishii and T. Fukuda. Speaker independent acoustic modeling using speaker normalization. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1, pages 97–100 vol.1, May 1998. doi: 10.1109/ICASSP.1998.674376.

[8] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2017. Draft.

[9] GrÃ©goire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-TÃ¼r, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:530–539, 2015.

[10] Ruslan Mitkov. *The Oxford Handbook of Computational Linguistics*. Oxford University Press Inc., New York, 1st edition, 2003. ISBN 9780199276349.

[11] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur. Spoken language understanding. *IEEE Signal Processing Magazine*, 25(3):50–58, May 2008. ISSN 1053-5888. doi: 1DOI10.1002/ecjc.20207.

[12] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986.

[13] Microsoft Research. Automatic speech recognition - an overview. URL https://www.youtube.com/watch?v=q67z7PTGRi8&t=1558s.

[14] Sharada C. Sajjan. Speech recognition using monophone and triphone based continuous density hidden markov models. 2015.

[15] G. Tur, D. Hakkani-TÃ¼r, and L. Heck. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24, Dec 2010. doi: 10.1109/SLT.2010.5700816.

[16] A. M. TURING. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 1950. doi: 10.1093/mind/LIX.236.433. URL http://dx.doi.org/10.1093/mind/LIX.236.433.

[17] Stevenn Volant, Caroline BÃ©rard, Marie-Laure Martin-Magniette, and StÃ©phane Robin. Hidden markov models with mixtures as emission distributions. 24, 06 2012.

[18] Ye-Yi Wang, Li Deng, and A. Acero. Spoken language understanding. *IEEE Signal Processing Magazine*, 22(5):16–31, Sept 2005. ISSN 1053-5888. doi: 10.1109/MSP.2005.1511821.

[19] Ye-Yi Wang, Li Deng, and Alex Acero. *Semantic Frame Based Spoken Language Understanding*, pages 35–80. Wiley, January 2011. URL https://www.microsoft.com/en-us/research/publication/semantic-frame-based-spoken-language-understanding/.

[20] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. Interspeech, August 2013. URL https://www.microsoft.com/en-us/research/publication/recurrent-neural-networks-for-language-understanding/.