

# Designing a scalable Conversational Interfaces

Cristian Gutierrez<sup>1§</sup>, Darshi Kasondra<sup>1§</sup>, Archana Ravi<sup>1§</sup>, Mounica Dingari<sup>1§</sup>

<sup>1</sup> California State University of Northridge

Additional work related to this paper can be found at <https://github.com/thehamop1/AdvSoftwareEngineeringProject>;

This version dated May 9, 2021

**Abstract** The way we use interact with systems is constantly evolving with new advances in technology. As business continues to need flexible user interfaces many of them are implementing conversational interfaces in order to fulfill customer needs without having to require users to learn how to use an entirely separate application in order to make transactions. Modern conversational interfaces are fairly powerful with recent advancements in natural language processing, integration with mobile assistants, and flexible frameworks. In this paper we provide background to some of the concepts of conversational interfaces, examining the current state of commercially available solutions, and provide a reference architecture for implementing a system centered around a conversational interface.

**\*For correspondence:**

[cristian.gutierrez.56@my.csun.edu](mailto:cristian.gutierrez.56@my.csun.edu) (CG)

**Present address:** <sup>§</sup>Department of Engineering and Computer Science, California State University of Northridge, United States of America

## 1 Introduction

### 1.1 Previous Work, Methods, Procedures

There is a growing need for approachable user interfaces as more interactions become digital. For example banking, transactions, and flights are all largely growing to be digital. In order to easily accessed and usable for all types of users conversational interfaces are the most ideal due to their low learning curve. If a user can use their native language in order to complete actions on any given system the need for 24/7 support or complex user interfaces become obsolete. This allows business to keep user satisfaction high while keeping costs low. Additionally it provides several benefits to users as they are able to access data from large databases easily, complete transactions even if their not so tech savvy, have multilingual support, and have 24/7 support.

### 1.2 Previous Work, Methods, Procedures

The development and interest of conversational interfaces has been a subject of interest since the 1970's. Early examples of these early chatbots include ELIZA, ALICE, and PARRY. Many of these early chatbots worked with the use of simple pattern matching. This simple regular expression based matching was combined with a tree design for controlling the flow of conversations. One of the major drawbacks of this naive form of design was the frequent matching of user utterances with conversation points that happened further up the tree. This would often lead to looping conversations. Researchers at the time had to develop markup based languages such as AIML in order to develop expert systems. These large complex forms of nested databases had to be constantly maintained in order to add new features to these chatbots. With the modern development of machine learning algorithms in order to parse and extract meaning from user utterances. Many com-

mercial solutions for developing conversational interfaces are now widely available and come with many integrations for various platforms.

### 1.3 Background

There are some basic concepts that are universal to most conversational interface platforms. The first is the user utterance which can either come in the form of a text entry or speech with the use of a microphone. Additional signal processing is required to transcribe the audio signals into text. This text is usually normalized where all text has its punctuation removed and all letters are moved into the same letter case. Next depending on the platform various machine learning algorithms are used in order to extract certain key tokens from the string. The most intent which is the objective of the user. For example the utterance "What is the weather in Los Angeles" would have the intent of weather. This is a topic that the application would have to be designed to respond to. Next would be entities which are certain tokens the application will use to complete requests once an intent has been deduced.