Methodology and Algorithms for Urdu Language Processing in a Conversational Agent

Mohammed Kaleem

A thesis submitted in partial fulfilment of the requirements of the Manchester Metropolitan University for the degree of Doctor of Philosophy

Manchester Metropolitan University

February 2015

Abstract

This thesis presents the research and development of a novel text based goal-orientated conversational agent (CA) for the Urdu language called UMAIR (Urdu Machine for Artificially Intelligent Recourse). A CA is a computer program that emulates a human in order to facilitate a conversation with the user. The aim is investigate the Urdu language and its lexical and grammatical features in order to, design a novel engine to handle the language unique features of Urdu. The weakness in current Conversational Agent (CA) engines is that they are not suited to be implemented in other languages which have grammar rules and structure totally different to English. From a historical perspective CA's including the design of scripting engines, scripting methodologies, resources and implementation procedures have been implemented for the most part in English and other Western languages (i.e. German and Spanish). The development of an Urdu conversational agent has required the research and development of new CA framework which incorporates methodologies and components in order overcome the language unique features of Urdu such as free word order, inconsistent use of space, diacritical marks and spelling. The new CA framework was utilised to implement UMAIR. UMAIR is a customer service agent for National Database and Registration Authority (NADRA) designed to answer user queries related to ID card and Passport applications. UMAIR is able to answer user queries related to the domain through discourse with the user by leading the conversation using questions and offering appropriate advice with the intention of leading the discourse to a pre-determined goal. The research and development of UMAIR led to the creation of several novel CA components, namely a new rule based Urdu CA engine which combines pattern matching and sentence/string similarity techniques along with new algorithms to process user utterances. Furthermore, a CA evaluation framework has been researched and tested which addresses the gap in research to develop the evaluation of natural language systems in general. Empirical end user evaluation has validated the new algorithms and components implemented in UMAIR. The results show that UMAIR is effective as an Urdu CA, with the majority of conversations leading to the goal of the conversation. Moreover the results also revealed that the components of the framework work well to mitigate the challenges of free word order and inconsistent word segmentation.

Contents

List of Figures

List of Equations

Dedication

This thesis is dedicated to my mother and father. Both of whom are like a candle – It consumes itself to light the way for others.

Acknowledgements

بِسْمِ اللَّـهِ الرَّحْمَـٰنِ الرَّحِيمِ

In the name of Allah, the Entirely Merciful, the Especially Merciful.

وَاشْكُرُوا لِلَّـهِ إِن كُنتُمْ إِيَّاهُ تَعْبُدُونَ

And be grateful to Allah if it is [indeed] Him that you worship. [2:172]

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا

For indeed, with hardship [will be] ease. [94:5]

In my limited vernacular I fall short to convey my thanks and sincere gratitude to Almighty Allah. I pray that Allah accepts my thanks for bestowing upon me truly countless blessings. I thank Allah for granting me with all the knowledge and abilities I possess and for the guidance I have received throughout my life. Indeed everything is from Allah and I am eternally grateful.

رَبَّنَا تَقَبَّلْ مِنَّا إِنَّكَ أَنْتَ السَّمِيعُ العَلِيمُ

Our Lord! Accept (this service) from us: For Thou art the All-Hearing, the All-knowing [2:127]

Two manifestations of my blessings from Allah are my mother (Bismillah Bibi) and father (Mohammed Naeem). Without their support and love I would not have been able to achieve a single thing in my life. Everything I am today is due to them and the sacrifices they have made for me and my brother and sister throughout their lives'. I want to thank my brother (Mohammed Aleem) and sister (Sarwat Bismillah) for their constant love, patience and support.

I would like to extend my sincere heartfelt gratitude to my supervisors Dr James O'Shea and Dr Keeley Crockett. I believe it was their unrelenting support, encouragement, expertise, guidance, time and belief in me that made me complete this PhD. It has truly been an honour and a pleasure to have worked alongside such people, who not only helped me through this PhD, but were also there whenever I needed them in any capacity. I am lucky to have had such great supervisors who made my PhD journey possible and picked me up and showed me the way during times I felt down and dejected. I learned a lot about myself and about life in general during this time.

To my childhood friends Asif Naseem and Arfan Ahmed thanks boys, I appreciate all the support and encouragement. I would also like to thank Hina Safarish Gill, for all the help and support with the Urdu language. I would like to extend a special thanks to Huda Aldossary, for sparking my academic journey and for being such a genuine friend. Special thanks to Omar Alobadi, thank you brother for all the help and support and the long discussions.

Last but by no means least, the person that I will be forever indebted to is Dr Shaikha Abdulatef AlSanad. Shaikha, I do not know how to thank you. I am lucky to have met someone like you, and I want to express my immense appreciation for everything you did for me and all the things you taught me. Thank you for the help, support and advice. You stood by me when I felt alone, and guided me when I was lost. Meeting you was a blessing. Thank you for being patient with me, during the days I was stressed/disheartened. Meeting you was one of the best consequences of this journey. I can say with conviction you will be my best friend for life, regardless of the geographical distance between us.

List of Publications

Kaleem, M., O'Shea, J. D., & Crockett, K. A. (2014, September). Word order variation and string similarity algorithm to reduce pattern scripting in pattern matching conversational agents. In Computational Intelligence (UKCI), 2014 14th UK Workshop on (pp. 1-8). IEEE.

Kaleem, M., K. A. Crockett and J. D. O'Shea (2014). Development of UMAIR the Urdu Conversational Agent for Customer Service. Proceedings of the World Congress on Engineering, IEANG 2014, London, July 2014, in press. (Awarded Best Student Paper Award of the 2014 International Conference of Computational Intelligence and Intelligent Systems)

Kaleem, M., K. A. Crockett and J. D. O'Shea (2014). UMAIR the Urdu Conversational Agent. Manchester Metropolitan University Research Day (July 2014).

List of Abbreviations

AI	Artificial Intelligence
AIML	Artificial Intelligence Mark-up Language
CA	Conversational Agent
CM	Conversation Manager
ECA	Embodied Conversational Agents
FAQ	Frequently Asked Question
GQM	Goal Question Metric
GUI	Graphical User Interface
IVA	Intelligent Virtual Agents
KB	Knowledge Base
NADRA	National Database and Registration Authority
NLP	Natural Language Processing
PARADISE	Paradigm for Dialogue System Evaluation
PM	Pattern Matching
STS	Short Text Similarity
UCA	Urdu Conversational Agent
UMAIR	Urdu Machine for Artificially Intelligent Recourse
WOW	Word Order Wizard
WOZ	Wizard of Oz
XML	Extensible Mark-up Language

Chapter 1 - Introduction

This thesis outlines a research endeavour undertaken to investigate whether a functional and effective Conversational Agent (CA) can be implemented for the Urdu language. The research entails thorough research in to the inner working of conversational agents, as well as the grammatical and morphological nature of the Urdu language and the inherent challenges that come with implementing the Urdu language in a CA. The research has led to the development of UMAIR (Urdu Machine for Artificially Intelligent Recourse). The architecture of UMAIR encompasses several new components that are specifically designed to handle the unique challenges of the Urdu language. This chapter provides the context and motivations behind this research, along with a short summary of the research contributions and a brief outline of the thesis structure.

1.1 Research Aims and Objectives

The primary focus of this research endeavour is to answer the research questions by testing hypotheses related to the research question. This will entail research in to CA's and the components they comprise of, as well as investigate the Urdu language and the unique features of the language with the aim of, designing a novel CA engine to handle the language unique features of Urdu. The project objectives are to use the research to design and implement a functional Urdu CA as a proof of concept to demonstrate the novel algorithms and components developed. The prototype CA should allow discourse with users, and provide some assistant within a selected problem domain.

1.1.1 Research Question

The primary question for this research is:

- Can the Urdu language be implemented in a CA to produce an effective, functional CA?

1.1.2 Aim

The primary aim of this research is to investigate appropriate methodologies to design and implement a novel Urdu Conversational Agent (UCA) architecture and an associated Urdu scripting language. The UCA architecture will consist of a novel Pattern Matching (PM) engine that is designed to handle the unique features of Urdu, and a new scripting language will be devised to deal with the shortcomings of existing English scripting methodologies. The effectiveness of the UCA will be evaluated through the development of a customer service orientated UCA for an organisation, that enables human participants to converse and discuss their queries with the agent and in turn receive directions as to the best course of action to solve their query.

1.1.3 Objectives

In order to answer the research questions/test the hypothesis the following objectives must be achieved:

(1) Investigate and evaluate existing CA scripting methodologies and engines to formulate a suitable implementation method for Urdu.

(2) Research and analyse the grammar, features and structure of the Urdu language along with existing methodologies of CA development, the techniques used to implement CA's. Formulate how these techniques can be used to extract responses from Urdu text. Subsequently, design a novel UCA engine architecture and scripting language for the implementation of a prototype UCA.

(3) Investigate knowledge engineering techniques to create a domain specific knowledge base and implement a new suitable knowledge base for the selected domain based on the results of the investigation.

(4) Implement the UCA, using the knowledge base developed in objective 3 and the architecture and the scripting language in objective 2 (based on findings of the findings of the language analysis in objective 2).

(5) Evaluate the final UCA by its ability to handle the Urdu language, as well as qualitative and quantitative end user evaluation through an appropriate evaluation framework.

Figure 1 outlines the research objectives, and where in this thesis it is addressed and situated.

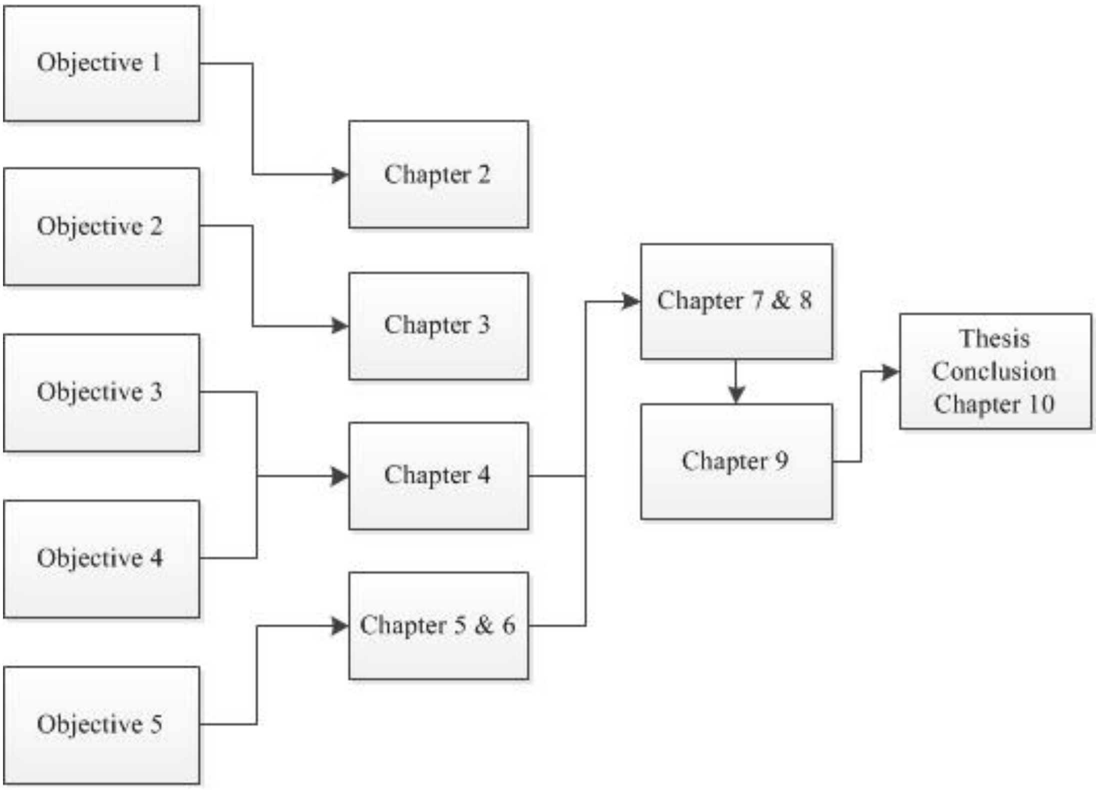


Figure 1 – Outline of objectives and where in the thesis they are addressed

1.2 Background and Motivation

A conversational agent is an autonomous software application design to converse with a user through natural language to provide instruction or advice related to a specific domain. In today's increasingly complex business environment, organisations face additional pressures regarding cost reduction, engagement scope, and attention to quality (Pickard et al., 2013). With this in mind, one of the most important emerging applications of CA's is online customer self-service, providing the user with the kind of services that would come from a knowledgeable or experienced human (O'Shea et al., 2008).

Urdu is the national language of Pakistan and one of the state languages of India and has more than 60 million first language speakers and more than 100 million total speakers in more than 20 countries (Gordon, 2005). Urdu script is written from right to left similar to other Semitic languages such as Arabic (Hardie, 2003).

In 2008 Pakistan was hit by the worst floods in its history, in light of this natural disaster a relief website was set up in English to give vital information about help, rescue efforts and shelter to those affected by the floods. The relief and recovery process is critical in nature, and needs to be made as efficient as possible. Information and Communication Technology (ICT) can play a key role in facilitating this process. However, the website proved to be quite ineffective in terms of dissemination of vital information until it was translated into Urdu (Sarfraz et al., 2010).

The traditional language for deployment of ICT solutions has been English, but it is evident that in order to reach the masses, the language medium needs to be one that is understood by the masses. Sriramesh et al. (2007) have stated that stress should be laid especially on the use of local languages given the fact that English is understood by only a small fraction of the population in Pakistan. This notion is supported by Sarfraz et al. (2010) who states given the low English literacy levels in Pakistan, it is evident that information disseminated in English will have a smaller audience and therefore a smaller impact.

Consequently, to make information accessible on a wider scale, in particular, to the large portion of the population that is not literate in English, it has to be localized into Urdu.

It is apparent that the web is playing a pivotal role in bringing information to the populations around the world (Sarfraz et al., 2011). Thus information available in localized contexts is more relevant to speakers of different languages; this is one of the drivers of this research. After several years of research and development activities CAs in English, European and East Asian languages CA's have become a popular area of research. But unfortunately South Asian Languages especially Urdu have received less attention (Anwar et al., 2006).

1.3 Research contributions

The most significant contributions of this work are:

- Proof of concept Urdu CA that demonstrates that it is indeed possible to implement a CA in the Urdu language that is able to mitigate and handle the language unique challenges of Urdu.

- A new Urdu scripting methodology and language that is designed specifically to allow fine control of the language unique features of Urdu during the scripting process.
- The WOW and word segmentation algorithms that mitigate the challenges of free word order and inconsistent word spacing that are features present in Urdu as well as other Eastern and South Asian languages. The algorithms reduce the scripting effort by processing the free word order and inconsistent word segmentation, therefore reducing the burden on the scripter to anticipate all possible variations of an utterance during the scripting process, saving substantial time and effort.
- An architecture for creating an Urdu Conversational Agent and a generic development methodology for creating conversational agents for resource poor languages.
- A new conversational agent evaluation framework has been developed and tested, which can be utilised to evaluate conversational agents from the objective and subjective perspective.
- A functional final prototype Urdu conversational agent and the results of two empirical evaluation experiments which validate the generic methodologies and architecture components.

1.4 Thesis Structure

The research conducted is presented in this thesis over ten chapters. The background review of existing literature and the current state of research related to the nature of this research is detailed over two chapters. Chapter two introduces and details the concept of conversational agents, the historical aspects of the field and the current state of the research, this is followed by a review of some existing CA's and the limitations of current CA architecture components. Chapter three provides an in depth overview of the grammatical and morphological nature of the Urdu language, along with the language unique challenges of the Urdu language.

Chapter four details the development process adopted to implement an Urdu CA. The development process is split in to four phase methodology, each of the phases of the methodology is explained in detail. Phase one and two form the methodology for implementing an Urdu CA. Phase one is the creation of a CA knowledge base. Phase two details the creation of the Urdu CA architecture components in depth and algorithms deployed to mitigate the challenges of the Urdu language. Phase three outlines the creation of UMAIR the Urdu conversational agent using methodology devised in phase one and two. Phase four outlines the evaluation of the developed Urdu CA (UMAIR).

Chapter five of the thesis presents the evaluation methodology and results of the empirical experiments carried out to test the research question/hypothesis from both objective and subjective aspects. The experiments involved end user participants who interacted with UMAIR and filled out questionnaires to rate different aspects of their experience with UMAIR. A Wizard of Oz (Schlögl et al., 2014) experiment was also carried out in order to test if UMAIR was comparable to a human in terms of objective task completion. This was followed by an analysis of the participant’s dialogue captured during their interaction with UMAIR and the WOZ in order to gauge the differences between UMAIR and WOZ.

Chapter six presents the results and a detailed discussion of the results of the experiments conducted to evaluate the first UMAIR prototype. The chapter highlights the aspects of the architecture that needed to be strengthened in order to increase the effectiveness, accuracy and robustness of UMAIR, such as the inconsistent word segmentation issue and spelling mistakes made by the users during their interaction with UMAIR.

Following the evaluation experiments, chapter seven illustrates the further research and development undertaken to strengthen UMAIR’s architecture to address the shortcomings and weaknesses highlighted through the first evaluation. This chapter details the amendments made to the existing architecture components as well as the new components added to the architecture that were researched and developed to overcome the shortcomings of the first prototype revealed through the first evaluation.

Chapter eight describes the evaluation methodology and the results of the experiments carried out to test the effectiveness of the second UMAIR prototype to determine whether the updates and additional components addressed the weaknesses found during the first evaluation.

Chapter nine provides a discussion related to the findings of the second evaluation and how the results impact the overall effectiveness of the second prototype compared to the first prototype. The discussion also outlines the results of the second evaluation and their impact on concluding the research hypothesis.

Chapter ten outlines the conclusions drawn from the research findings and discussion, the main contributions of the research as well as providing recommendations for future research avenues that could be followed.

Chapter 2 - Conversational Agents

2.1 Introduction

Natural language communication with a computer has been a goal in the field of artificial intelligence for many decades inspired by the Turing Test (Turing, 1950). The Turing test was devised to evaluate whether a computer program could convince a judge that they were actually conversing with a human. Early attempts at passing the Turing test involved computer programs called chatterbots, that employed tricks during the conversation to create the illusion of intelligence, but in actual fact had no intelligence actually programmed in to the system (Weizenbaum, 1966). More recent developments in the field have produced artificially intelligent Conversational Agents (CAs). CA's facilitate communication between humans and computer using natural language (O'Shea et al., 2014) that are able to mimic human experts to offer domain specific advice and information to the user (O'Shea et al., 2011) in order for them to reach some pre-defined goal for example technical advice related to some product or device.

The creation of a new UCA is based on principles adopted from research in several key areas, specifically the Urdu language, conversational agent architectures and goal orientated CA's. This chapter reviews the literature to outline the different approaches/methodologies to implementing CA's. CA's are defined with relation to their functionality and examples are reviewed and discussed. Subsequent sections explore and outline scenarios where CA's have been applied and the two main types of CA's in terms of functionality (i.e. linguistic text based CA's and embodied CA's). CA development is thoroughly explored and CA architecture components are outlined in terms of their functionality and contribution to the overall architecture. Subsequent to this CA knowledge bases are investigated, along with knowledge base development techniques and their associated shortcomings. Finally, CA evaluation methodologies are detailed and a possible new framework/approach for the evaluation of CA's is proposed and outlined.

2.2 Conversational Agents

The term "Conversational Agent" (CA) is interpreted in different ways by different researchers; however the essence of CAs is natural language dialogue between the human and an application running on a computer (O'Shea et al., 2011). Rubin et al. (2010), define them as a natural language interaction interface designed to simulate conversation with a real person. According to Alobaidi et al. (2013) a CA is an agent which uses natural language dialogue to communicate with users. Lester et al. (2004) posit that CA's exploit natural language technologies to engage users in text-based information-seeking and task-oriented dialogs for a broad range of applications.

Conversational agents are representative intelligent agents that are able to respond to user requests and queries in an intelligent way (with natural language dialogue). They can understand the intention of users through conversation normally through a text based interface, after understanding, they are able to offer an appropriate service or advice. A CA also has the ability to reason and pursue a course of action based on its interactions with humans and other agents (Crockett et al., 2011). There are two distinct categories of CAs, 'Embodied CAs' and 'Linguistic CAs' (Mairesse et al., 2007, O'Shea et al., 2014). Embodied CAs are animated anthropomorphic interface agents, that can communicate with a user using verbal and paralinguistic methods for example embodied CAs often possess an animated humanoid body and exhibit attributes such as facial expressions and movement of eye gaze (Malatesta et al., 2009, O'Shea et al., 2014). While Linguistic CA's handle conversation in written or spoken forms (Cassell, 2000a). For the purpose of the research carried out in this project, the main focus will be on linguistic CA's as this research in is an initial step into the creation of a linguistic Urdu language CA.

One of the earliest examples of a CA developed was ELIZA (Weizenbaum, 1966). ELIZA is a Chatbot capable of creating the illusion that the agent was actually listening and understanding the user's utterances and providing intelligent responses posed as questions emulating a Rogerian psychotherapist (Rzepka and Araki, 2015), however it was just using simple pattern matching techniques that worked by simply parsing and recomposing key words based on the user input to formulate responses. ELIZA's main trick was to use questions to draw a conversation out of the user, the main criticism ELIZA and other Chatbot applications faced was the program's lack of intelligence and context awareness that could influence, track and direct the conversation (Crockett et al., 2011).

As the field of CA's advanced, ALICE (Artificial Linguistic Intelligent Computer Entity) was produced. The knowledge base for ALICE is stored in AIML (Artificial Intelligent Markup Language) files. Fundamentally AIML is a pattern matching scripting language derived from Extensible Markup Language (XML) and used symbolic reduction to parse user utterances and generate responses. What is considered to be the brain of ALICE is made up of around 41,000 elements called categories. Each category combines a question and answer, or stimulus and response, known as the "pattern" and "template" respectively. The AIML software stores the patterns in a tree like structure and is managed by an object called the graphmaster, implementing a pattern storage and matching algorithm (Wallace, 2009). In ALICE, the AIML technology was responsible for pattern matching and to relate a user input with a response in the chatterbot's Knowledge Base (KB) (Marietto et al., 2013). This is achieved through the process of symbolic reduction which broke the user input down to its constituent parts in order to find matches to the patterns. In essence the ALICE engine is a more refined version of the simpler engine used in ELIZA (Shawar and Atwell, 2002) however it still lacked the sophistication of more recent engines.

An example of a recent CA is InfoChat (Michie and Sammut, 2001). InfoChat introduced some new approaches to CA development and scripting in an effort to add some artificial intelligence to the discussion between the user and the CA in order for the conversation to reach a goal. InfoChat implements a pattern matching approach using a sophisticated scripting language known as Pattern Script. InfoChat scripting language is a rule-based language, which depends on a rule based structure to handle the expected conversation, However, it also uses the concept of "spreading activation", which strengthens or inhibits rule firing based on conversation history adding a level of awareness to the system. Furthermore, InfoChat introduced a more sophisticated technique of determining the similarity of user utterances and scripted patterns, which is calculated through several parameters such as activation level and pattern strength. The majority of CA's utilise Pattern Matching (PM) and language scripting techniques within their engines. Within CA's Scripts are usually organized into contexts consisting of a number of hierarchically arranged rules (Sammut, 2001). Scripts are typically scripted using rules as shown in Figure 2.

```
<Rule_01>
a:0.5
p:50 *<confused-0>*
p:50 *<confusing-0>*
p:50 *<sure-neg-0>*
p:50 *<sure-neg-1>*
p:50 *help*
p:50 *not *<understand-0>*
r:What can I do for you?
```

Figure 2 – Example scripted rule from InfoChat

Each rule possesses a list of structural patterns of sentences and an associated User input is matched against each pattern through an engine with the intention of finding a match (O'Shea et al., 2014). The scripts are used to structure and organise the knowledge base. The knowledge is broken down into contexts and each context consists of rules which in turn have patterns to represent them. The rule also has an associated response which is conveyed back to the user when that rule is invoked through pattern matching the user utterance.

Nevertheless the disadvantages of these earlier systems mentioned (ELIZA, ALICE) was that their knowledge bases are very general, they tended to have a general breadth of knowledge but no depth allowing for shallow, general conversations only, exhibiting little or no intelligence. Furthermore, the vast majority of conversational agent systems have been developed for English, therefore are not suitable for Urdu and other eastern languages due to the difference in grammar and written system (explored in depth in chapter three). The key features of a conversational agent can be summarised as:

- A CA is a computer program that facilitates natural language dialog with a computer.
- A CA enables autonomous 24 hour information access to users.
- A CA system architecture has many different components, common components are:
    - **A knowledge base** with provides it with knowledge related to a certain domain (Alobaidi et al., 2013, O'Shea et al., 2014).

- **An engine** that processes the user utterances against patterns stored in the knowledge base with the purpose of finding a match and then delivering a response back to the user (Kaleem et al., 2014a, Latham et al., 2014, Latham et al., 2010a, O'Shea et al., 2011, O'Shea et al., 2010, Kaleem et al., 2014b).
- **Memory** which can be short or long term which allows the system the ability to remember conversation related information (Richards and Bransky, 2014, O'Shea et al., 2014)**.**
- **A user interface** (UI) that enables the human user to input text in to the system (Cassell, 2000b, Nunamaker et al., 2011, O'Shea et al., 2011).

The following sections will delve deep into these features with the intention of outlining the contribution each of these components and features have in relation to the CA functionality.

2.3 Applications of CAs

There is a high variety of applications in which conversational agents can be used, one of the most widespread of which is information retrieval (Griol et al., 2013). CA's have been deployed on retail websites (Etemad-Sajadi, 2014, Kulms et al., 2014), where they respond to customers' inquiries about products and services. CA's associated with financial services' websites answer questions about account balances and provide portfolio information. CA's for entertainment are deployed in games to engage players in situated dialogs about the game-world events (Lester et al., 2004). Pedagogical CA's assist students by providing problem- solving advice as they learn (Hayashi, 2013, Alobaidi et al., 2013). A more recent 'main stream commercial' application of intelligent agents has been the virtual personal assistant, popular examples of which are Apple Inc.'s "Siri" (Apple, 2014), Microsoft's 'Cortana' (Microsoft, 2014) and Google's 'OK Google' (Google, 2014). These personal assistants are all voice based conversational agents, however the core functionality remains similar as traditional CA approaches. The user speaks a command and the application synthesises the speech in to text which is then processed by an engine to generate an appropriate response (Bellegarda, 2014).

It is apparent that CA's are starting to play a more prominent role in everyday applications and in one particular sector of applications, that being enterprise software (Lester et al., 2004). In recent years, the demand for more cost-effective solutions to the customer service problem has increased dramatically as companies are looking to save money wherever possible in response to the global economic downturn. Implementing automated solutions such as CA's can significantly reduce the high customer service budgets that companies have devoted to training and labour costs. Through exploiting the enabling technologies of the Web and advances computational linguistics, conversational agents offer companies the ability to provide customer service much more economically than with traditional methods (Silvervarg and Jönsson, 2011).

Effective communication is principal for a wide range of tasks in enterprise. Communication comprising information-seeking and task/goal-oriented dialogues is central to many major families of business applications which have seen CA's implemented to handle varying tasks:

1. Customer service: Responding to customers' general questions about products and services, e.g., answering questions about problems/queries in a given domain (Rubin et al., 2010).
2. Help desk: Responding to internal employee questions, e.g., responding to HR questions (Lester et al., 2004).
3. Website navigation: Guiding customers to relevant portions of complex websites. A "Website concierge" is invaluable in helping people determine where information or services reside on a company's website (Shimazu, 2002).
4. Guided selling: Providing answers and guidance in the sales process, particularly for complex products being sold to novice customers (Keeling et al., 2004).
5. Technical support: Responding to technical problems, such as diagnosing a problem with a device (O'Shea et al., 2011).
6. Education – Conversational intelligent tutoring systems (Alobaidi et al., 2013, Latham et al., 2014).
7. HR Bully and Harassment Help System (Latham et al., 2010a)

In customer facing deployments, conversational agents interact directly with customers to help them obtain answers to their questions (Lester et al., 2004). In this type of application CA's have been very successful and users have expressed their appreciation of the systems. This is evident in the evaluation results of the HR Bully and Harassment Help System developed by Latham et al. (2010a), who's findings include among others that the vast majority users of their system were "able to find the information they sought without difficulty".

2.4 Embodied CAs

Embodied conversational agents (ECA) are computer-generated characters that demonstrate many of the same properties as humans in face-to-face conversation, including the ability to produce and respond to verbal and non-verbal communication (Cassell, 2000a, Derrick and Ligon, 2014). According to Derrick and Ligon (2014) ECA's are human-like renderings, often coupled with environmental sensors that interface with people in natural ways. An Embodied CA is regarded as a multimodal interface which displays a face, hand gestures, expressions etc., in order to interact with a human (or representation of a human in a computer environment) and a dialogue system where both verbal and nonverbal devices advance and regulate the dialogue between the user and the computer (Cassell et al., 2001, Boisseleau et al., 2014).

According to O'Shea et al. (2011) the extent of embodiment can vary considerably. They suggest that embodiment in its simplest from involves a graphic representation of the agent which is capable of facial expressions, where the intention is to provide a generally heightened sense of realism/naturalness. An advanced implementation of embodiment would be an agent capable of simulating facial expressions and human like gestures (Babu et al., 2006).

Cassell et al. (2001) argue that embodiment can serve an even stronger function if system designers use actual human conversational protocols in the design of the interface. For example, communicative behaviours such as salutations and farewells, conversational turn-taking with interruptions, and describing objects using hand gestures are examples of protocols that all native speakers of a language already know how to perform and can thus be utilised in creating a more natural intelligent interface, embodiment is required for the successful implementation of CA's (Cassell et al., 2001). This notion is supported by O'Shea et al. (2011) who state that there is clear potential for embodiment to improve CA's, for example through disambiguating pronouns such as this and that using pointing gestures to provide visual clues and hints. The embodied character or visual representation of the CA should complement the UI and not look incongruous to the overall interface. Since this research is more focused on the development of a new Urdu CA engine and addressing the linguistic challenges of Urdu, the main focus will be the algorithms and methodologies required to achieve an effective UCA.

2.5 Goal Orientated Conversational Agents (GO-CA)

A Goal-Oriented CA (GO-CA) is a type of conversational agent which has a deep strategic purpose which enables it to direct a conversation to achieve a goal (O'Shea et al., 2011, Crockett et al., 2011). The predominant difference between traditional CA and GO-CA is that a GO-CA through the process of dialogue, captures appropriate attributes to model the particular problem experienced by the user in order identify the appropriate solution (O'Shea et al., 2011) and lead the discussion towards achieving the goal.

GO-CA's are designed to converse with humans through the use of natural language dialogue to achieve a specific task (Crockett et al., 2010, O'Shea et al., 2011). For example, identifying and selling a person a mortgage or providing guidance through an organisation's policies and procedures in plain English (Latham et al., 2010a). Traditionally, they utilise pattern matching algorithms to capture the values of specific attributes through text based discussion with a user. This is achieved through the use of scripts which contain sets of rules about the domain and a knowledge base to guide the conversation towards achieving a specific goal.

GO-CA's systems can provide anonymous, automated, interactive and consistent advice 24 hours a day in many different scenarios, including online customer self-service/assistance, providing the user with the kind of services that would come from a knowledgeable or experienced human (O'Shea et al., 2008) for example helpdesk/customer service agents that respond to customers' inquiries about products and services (Rubin et al., 2010).

Pedagogical conversational agents (also known as Intelligent Tutoring Systems) that assist students by providing problem- solving advice as they learn with the goal of delivering some learning based objective (Alobaidi et al., 2013, Latham et al., 2014).

Typically, the vast majority of GO-CA's to date have been deployed in the English language, and development in other languages is little to non-existent.

2.6 Conversational Agent Engines

CAs engines have been developed using many different techniques. The three main techniques are Natural Language Processing (NLP) Short Text Semantic Similarity (STSS), and Pattern Matching (PM). NLP, STSS and PM are approaches that differ from Machine Translation (MT), as the aim of machine translation is to translate text in one language to another. Whereas, the CA technique aims to process the text in order to understand it and formulate an appropriate response. In the development of sophisticated natural language processing systems, it is understood that a rich lexical knowledge base is at the heart of any intelligent system that attempts to go beyond the syntactic analysis of sentences (Ahmed and Hautli, 2011). A lexical resource such as a WordNet can shed light on the meaning of a sentence by providing information on the lexical semantics of the words in the sentence. However, these lexical resources face a serious drawback: their development is time-consuming, costly and requires trained linguists that are aware of the lexical variation of a language. The task becomes even harder when only few other resources for the language exist which are readily available and the possibilities for automatic acquisition of data are rather restricted. With research being mainly focused on European languages like English and German (Bender, 2009, Almarsoomi et al., 2012). This resource sparseness is a problem for Urdu, which is explored further in chapter 3 (Ahmed and Hautli, 2011).

NLP is an area of research that explores how computers can be used to understand and manipulate natural language text or speech to do useful things (Chowdhury, 2003). NLP assumes certain aspects for it to work effectively. The utterance is expected to be grammatically correct which usually it is not, incorrect sentences may be "repaired" but this adds computational overhead. Another point is that languages are very rich in form and structure, and contain ambiguities. A word can have more than one meaning (lexical ambiguity) or a sentence might have more than one structure (syntactic ambiguity/free word order), in light of this the NLP approach is not suitable to develop a CA in the Urdu language.

Another approach that is adopted in the development of CA's is the utilisation of Short Text Semantic Similarity (STSS) measures to gauge the similarity between short sentences (10 – 25 words longs) (O'Shea et al., 2008, O'Shea et al., 2014). Through employing sentence similarity measures, scripting can be reduced to a few prototype sentences (O'Shea et al., 2009). The similarity between short texts is computed through the use of a knowledge base such as the English WordNet or text corpora and an algorithm/measure that utilises the knowledge base resource to calculate the similarity between two texts of short length. However due to the lack of resources in Urdu such as an appropriate WordNet, lexicons, annotated electronic dictionaries, corpora and well-developed ontologies that describe relationships among words and entities in written text (Naseem and Hussain, 2007) NLP and STSS are not appropriate methods to develop a Urdu CA, and to date no Urdu STSS measures exist. It should be noted that work has begun on the development of an Urdu WordNet (Zafar et al., 2012). The work is still in very early stages and not developed enough to be deployed in a CA, because the current Urdu WordNet is a translated derivative of the Hindi WordNet therefore is incomplete and contains words that are not used in Urdu which in its current state of development makes it unsuitable to be use in a CA (Adeeba and Hussain, 2011).

The remaining technique known as Pattern Matching (PM) is one of the most ubiquitous and popular methods for building systems that appear to be able to conduct coherent, intelligent dialogues with users (Bickmore and Giorgino, 2006). Most text-based CA's adopt the pattern matching approach as it is currently the one that works best for extended dialogues (O'Shea, 2011). This notion is also supported by Allen et al. (2001) who state that pattern-matching techniques are used to great effect in dialogue systems. The PM approach aims to match the user utterance to a database of pre-scripted patterns, rather than trying to understand the utterance. Once a pattern is matched an appropriate response is delivered back to the user.

PM CA's use a pre-compiled repository of scripts, which are grouped into contexts (Illustrated in Figure 3). Each context is made up of a number of rules. Each rule consists of a number of patterns and a linked response which make up the CA's knowledge base (Kaleem et al., 2014b).

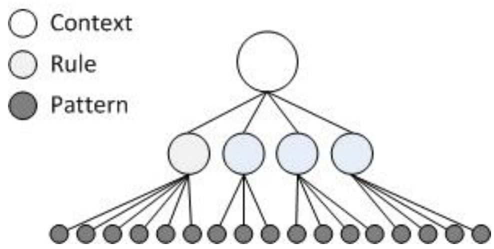


Figure 3 - Scripting hierarchy of a single context

Each rule is the sub-topic/context that relates to an attribute of the main context that a user utterance may be matched with. Each rule can have a number of different patterns that are used to match it with a user utterance. Patterns consist of a collection of words and wildcard symbols (e.g. *, $), wildcards are used within patterns to match any number of words, broadening the rules to match utterances containing specific key phrases. An example of a scripted rule is illustrated in Figure 4.

Context ID Card – Application Form
Rule – App_Form Pattern: * form do I need to for a new ID card Pattern: * which form shall I fill * ID card Pattern: * need a form a new ID card Pattern: * form to apply for a replacement ID card Response: To apply for a new ID card you need to fill a POC form.

Figure 4 - Example of a general scripted rule

An algorithm decides the best fitting rule to fire and deals with conflict resolution in situations where multiple rules fire, thus producing a CA response. PM is a suitable method for developing an Urdu CA as it does not require extensive lexical resources, or grammatically correct or complete user input to work. The PM approach has been used to create CA's in the Arabic language, which show promising results (Alobaidi et al., 2013, Hijjawi et al., 2014). Nonetheless due to the inherent difference between Urdu and Arabic (e.g. additional characters, word order) it not possible to use an Arabic CA engine to process Urdu text. Consequently, it is necessary to research and develop a PM engine specifically for the Urdu language.

However, there are some drawbacks of the PM approach which are the scripting process itself and the subsequent maintenance of the scripts. Traditional CA scripting requires the script writer to consider every permutation of a user utterance that a user may send as input (O'Shea, 2013). The PM approach requires precompiled scripts that define the conversation to be executed by a pattern-matching engine. Scripting is a time-consuming process. It is focused solely on the structural form of the sentence. This requires the anticipation of all possible user utterances, generation of word order permutations of the utterances and generalization of patterns through the replacement of selected terms by wild cards. The main disadvantage of pattern matching systems is the labour-intensive (and therefore costly) nature of their development (O'Shea et al., 2011).

Furthermore, modifications to rules containing the patterns can impact on the performance of other rules. Consequently the entire database of scripts has to be reassessed in order to maintain the integrity of the scripted rules and avoid rule clashes and misfiring rules. This is a high maintenance cost and almost impossible process. In addition, different script writers possess differing levels of ability and as such this can prove to be an exasperating task (O'Shea, 2013).

An example of a PM CA is InfoChat (Michie and Sammut, 2001). InfoChat implements a pattern matching approach using a sophisticated scripting language known as Pattern Script (Michie and Sammut, 2001). InfoChat scripting language is a rule-based language, using the type of rule structure shown in Figure 2 to handle the expected conversation. InfoChat was further developed by Convagent which did try and aim to automate the scripting process using a Goal-orientated tree tool. Semi automation was achieved through the tool but the problems of script development and maintenance still remained.

2.7 Conversation Agent Knowledge Base Development

According to Engelmore and Feigenbaum (1993) GO-CA's are expert systems which, contain two essential parts: the knowledge base; and the reasoning, or inference, engine. The knowledge base of such systems is comprised of both factual and heuristic/experiential knowledge. Factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field. Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely subjective. It is the knowledge of good practice, good judgment, and plausible reasoning in the field.

The knowledge representation process formalizes and organizes the knowledge. One widely used representation is the production rule, or simply rule (Engelmore and Feigenbaum, 1993). A rule consists of an IF part and a THEN part (also called a condition and an action). The IF part lists a set of conditions in some logical combination. The piece of knowledge represented by the production rule is relevant to the line of reasoning being developed if the IF part of the rule is satisfied; consequently, the THEN part can be concluded, or its problem-solving action taken. Expert systems whose knowledge is represented in rule form are called rule-based systems. (Engelmore and Feigenbaum, 1993, Buchanan and Shortliffe, 1984, Agbo-Ajala et al., 2014).

Historically the development of a Knowledge Base (KB) was seen as a transfer process of human knowledge into an implemented structured knowledge base. This transfer was based on the assumption that the knowledge which is required by the KB already exists and just has to be collected and implemented. Most often, the required knowledge is obtained by interviewing experts on how they solve specific tasks (Musen, 1993). Typically, this knowledge is implemented as production rules which are then executed by an associated rule interpreter/engine. More recently an overall consensus has emerged that the process of building a KB may be seen as a modelling activity. Constructing a KB means building a computer model with the aim of realising problem-solving capabilities comparable to a domain expert. It is not intended to create a cognitive adequate model, i.e. to simulate the cognitive processes of an expert in general, but to create a model which offers similar results in problem-solving for problems in the particular domain or area of concern (Morik, 1991, Studer et al., 1998).

This modelling view of the knowledge base building process has the following consequences:

- Like every model, a knowledge model is only an approximation of the reality. In principle, the modelling process is an on-going process, because it is a continuous activity with the aim of approximating the intended behaviour.

- The knowledge modelling process is a cyclic process. New observations and testing may lead to a refinement, modification or completion of the already built-up model. On the other hand, the model may guide the further acquisition of knowledge.
- The knowledge modelling process is dependent on the subjective interpretations of the knowledge engineer. Therefore, this process is never perfect, thus an evaluation of the model with respect to reality is indispensable for the creation of an adequate model. According to this feedback loop, the model must, therefore, be revisable and adaptable in every stage of the modelling process (Studer et al., 1998).

The initial phase of creating a KBS it knowledge extraction, where the knowledge required is collated and recorded so it can be stored and structured accordingly. According to O'Shea et al. (2011) knowledge about a domain is extracted from many different sources, including:

- Managers in the client organisation
- Practitioners in the client organisation who interact with the customers who will use the CA being developed
- Documented procedures of the client organisation (e.g. workflow charts)
- 3rd party websites (e.g. government legislation concerning the domain)
- Telephone logs of customer calls related to the domain.

CA's utilise structured knowledge bases in order to store knowledge such as conversation scripts, rules and responses specifically related to the domain implementation. The user utterances are matched to the scripts in the knowledge base that in turn fire rules that have responses associated with them. The architecture of a CA encompasses a KB that that is related with the agent's domain, examples include, sales (Bickmore and Cassell, 2005), debt advice (Crockett et al., 2009) or teaching the main principles of Islam (Alobaidi et al., 2013) and a dynamic discourse knowledge base that deals with what has already been said (i.e. memory) (Cassell, 2000b) .

2.8 CA's and Memory

To engage in any form of dialogue, an aspect of memory is essential. Human memories may be triggered through the use of clues, cue words and through the use of semantic relations. CA design must incorporate an aspect of memory simulation in order to develop a human-like dialogue. According to Baddeley (1999) an important requirement for successful human interaction is our ability to store, retain, recall and organise information. This ability is known as memory and it is essential to our basic functioning as human beings. Memory performs the important functions of learning, organising and remembering, forgetting, repression, storage and retrieval; all of which centre on information related to facts, details and events. For instance, our ability to remember time and events allows us to keep track of what we have done, and to make plans for the future.

Furthermore, memory plays an essential role in fostering trust between humans and without memory the notion of a companion with whom you share experiences would be meaningless. Memory has been studied extensively. Increasingly architectures, both agent and cognitive, include memory modules to ensure retention of relevant information. Memory will be particularly important for Intelligent Virtual Agents (IVA)/CA's that continue to be found useful after their novelty effect has worn off (Kasap and Magnenat-Thalmann, 2012).

Kasap et al. (2009), have researched memory in a pedagogical virtual agent they designed called 'Eva'. They state that memory has typically been implemented to address the issue of how agents remember information from one interaction to another. However memory is essential for agents to effectively carry out the role for which they are designed. An example for a pedagogical agent is it needs to remember past lessons held with a student as in the case of Eva, who uses a memory-based emotion model and memories of past interactions (i.e., episodic memory) to build interpersonal relationships with users. The inclusion of memory models in Intelligent Virtual Agent's (IVA's) is similar to the inclusion of student models in intelligent tutoring systems these student models may contain learning achievements, preferences and learning styles.

Brom and Lukavský (2009) have also stressed the need for memory in agents. They state that it is necessary for agents to utilise memory for a broad range of tasks like debriefing, giving information, remembering the course of interactions, searching for objects, knowledge sharing and learning. It is to be noted that the overarching concept behind the creation of intelligent virtual agents/CA's is believability, where the primary goal is to produce agents that imitate human-like behaviour (Richards and Bransky, 2014). A CA that can exhibit a human-form of memory can develop a more meaningful relationship with its user resulting in a broad range of dialogue opportunities. This leads the way for a truly intelligent agent. The inclusion of memory to a CA adds self-awareness, character and intelligence (O'Shea, 2011).

In CA's short term memory relates the system remembering conversation related variables such as the users name and the context of the discussion. Long term memory is the long term storage of the captured variables in a database or other medium which can be utilised at a later date in order for the system to simulate recall of previous knowledge and discussions with returning users (Richards and Bransky, 2014).

2.9 Conversational Agent Evaluation

In the IEEE Glossary of Software System Engineering Terminology (IEEE, 2000), quality is defined as the degree to which a system, a component, or a process meets customer or user needs or expectations. According to (Roy and Graham, 2008), the quality of software is measured primarily against the degree to which requirements, such as correctness, reliability and usability are met. The factors that affect quality are termed as quality attributes. There are different categorisations of quality attributes. They further state that quality attributes can be categorized into two broad groups: attributes that can be directly measured (e.g. performance) and attributes that can be indirectly measured (e.g. usability). These attributes can be translated into objective and subjective metrics respectively.

In order to build a conversational system, data is needed on how users behave and their perceptions when interacting with the system (Skantze and Hjalmarsson, 2013). According to Martinez et al. (2008), it is quite difficult to evaluate dialogue systems. In addition to the lack of evaluation standards within the dialogue community, at the same time, it is difficult to find performance figures from real world applications that can be extrapolated to other systems or be worldwide accepted, as all of them are directly related to one specific dialogue system. An early example of evaluating the success of dialog based software is the Turing test. The Turing test (Turing, 1950) was primarily aimed at making a human believe that they were speaking to another human, when in fact they were speaking to a computer program. This approach however is not suitable to gauge the effectiveness or usability of a goal orientated conversational agent as the intrinsic nature behind the two applications are completely different.

There is a general agreement on “usability” as the most important performance figure (Turunen et al., 2006, Walker et al., 2000) even more than others widely used like “naturalness” or “flexibility”. However functionality may be more important, but without usability the system will not get the chance to demonstrate functionality. Therefore, besides quality and efficiency metrics, automatically logged or computed, subjective tests have also been performed in order to assess the impact of the capabilities of the system on user satisfaction and to get a valuable insight on the shortcomings and advantages of the system (Martinez et al., 2008).

A substantial amount of work has been done on evaluating CA’s as a whole. The seminal work in this area was done by Walker et al. (1997) who created the PARADISE framework. An important feature of PARADISE is the application of linear regression for deriving abstract, indirect attributes such as user satisfaction in terms of directly measurable attributes (Fenton and Pfleeger, 1998). For determining the quality of Spoken Dialogue Systems, several aspects are of interest. Moller et al. (2009), presented a taxonomy of quality criteria. They describe quality as two separate issues consisting of Quality of Service (QoS) and Quality of Experience (QoE). Quality of Service describes objective criteria like dialogue duration or number of turns or utterances it takes to achieve the desired outcome. While these are well-defined items that can be determined easily, Quality of Experience, which describes the user experience with subjective criteria, is a more vague area and without a sound definition, e.g. User Satisfaction.

According to Silvervarg and Jönsson (2011), the evaluation of CA/dialogue systems is mainly done either by distributing a questionnaire to the users trying to the reveal their subjective assessment of using the dialogue system or by studying the resulting dialogue. Artstein et al. (2009), refer to this as “soft” numbers versus “hard” numbers and propose a “semi-formal” evaluation method combining the two evaluation methodologies. This notion is supported by more recent research conducted by Rauschenberger et al. (2013) who propose a framework to measure user experience and software quality in interactive software applications through User Evaluation Questionnaires (UEQ). They state that the evaluation of interactive software quality falls into two distinct categories, these being “pragmatic quality” and “hedonic quality”. Pragmatic quality relates to task orientated quality like task completion effectiveness and efficiency. Hedonic quality is related to non-task orientated aspects like aesthetic impressions and user stimulation. These two categories can be translated into objective measures and subjective measures respectively.

The general consensus among researchers in the field (Alobaidi et al., 2013, O’Shea et al., 2011, O'Shea et al., 2009) from the early days to the present day is that the effectiveness of a CA/Dialogue system should be evaluated through a combination of subjective and objective measures. This ensures that not only is the effectiveness of the CA’s functionality tested but the usability from the user perspective is also tested.

As there has been no formal development to the CA evaluation frameworks over the years, alternative approaches/evaluation frameworks that can be adopted are software evaluation frameworks that that are utilised to test new software applications in terms of functionality and usability (i.e. objective and subjective metrics).

2.9.1 Formulation of Evaluation Metrics

As with any engineering discipline, software development requires a measurement mechanism for feedback and evaluation. Measurement is an aid in answering a variety of questions associated with the enactment of any software. It allows the determination of the strengths and weaknesses of the current processes and allows us to evaluate the quality of specific processes and products (Van Solingen et al., 2002). A particular measurement/evaluation is useful only if it helps you to understand the underlying process or one of its resultant products. In turn, recognizing improvement of the process and products can occur only when the project has clearly defined goals for process and products. In other words, you cannot tell if you are going in the right direction until you determine your destination. (Fenton and Pfleeger, 1998).

According to (Fenton and Pfleeger, 1998) an evaluation strategy can be more successful if it is designed with the goals of the project in mind. One such strategy is the Goal Question Metric (GQM) approach, which is based upon the assumption that for an organization to measure in a focused way it must first identify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals (Van Solingen et al., 2002).

Thus it is important to make clear, at least in general terms, what informational needs the organization has, so that these needs for information can be quantified whenever possible, and the quantified information can be analysed as to whether or not the goals are achieved.

The GQM approach provides a framework involving three steps:

1. (**GOAL**) List the major goals of the development or maintenance project.
2. (**QUESTION**) Derive from each goal the questions that must be answered to determine if the goals are being met. Questions try to characterize the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint. Once the questions have been developed, the next step involves associating the question with appropriate metrics.
3. (**METRIC**) Decide what must be measured in order to be able to answer the questions adequately. A set of data is associated with every question in order to answer it in a quantitative way. The data can be
    - **Objective**: If they depend only on the object that is being measured and not on the viewpoint from which they are taken; e.g., number of versions of a document, staff hours spent on a task, size of a program.
    - **Subjective**: If they depend on both the object that is being measured and the viewpoint from which they are taken; e.g., readability of a text, level of user satisfaction.

(Fenton and Pfleeger, 1998, Van Solingen et al., 2002)

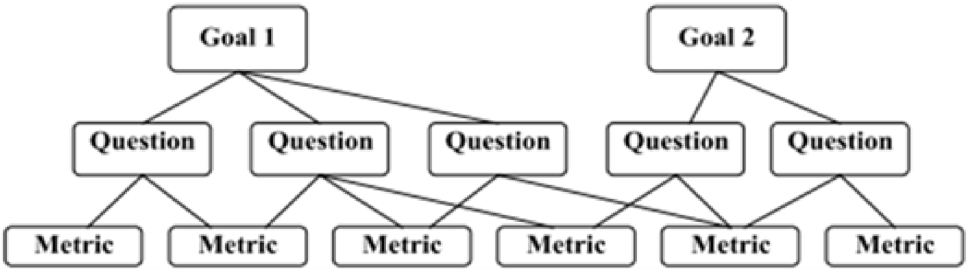


Figure 5 - GQM Model (Van Solingen et al., 2002)

A GQM model is a top down hierarchical model as illustrated in Figure 5, the top level starts with a goal (specifying purpose of measurement, object to be measured, issue to be measured, and viewpoint from which the measure is taken). The goal is refined into several questions that usually break down the issue into its major components. Each question is a metric, some of them objective, some of them subjective. The same metric can be used in order to answer different questions under the same goal (Van Solingen et al., 2002). Traditional CA evaluation methodologies all encompass objective metrics to evaluate the effectiveness of the developed system (Walker et al., 1997).

2.9.2 Subjective Evaluation Methodologies

Subjective aspects like user satisfaction are determined by using questionnaires (Hone and Graham, 2000, Silvervarg and Jönsson, 2011, Rauschenberger et al., 2013) the satisfaction ratings are applied either:

- by users during or right after the dialogue or
- by experts by analysing recorded dialogues

According to Brooke (1996) the usability of any tool or system has to be viewed in terms of the context in which it is used, and the degree of its appropriateness to that context. Accordingly user Satisfaction related to a CA is only possible by asking real users about interactions with the system (Ultes et al., 2013). Based on this notion it is proposed that the most efficient and effective method to gauge usability and end user satisfaction is to administer a questionnaire to the participants to obtain their individual views and opinions with regards to the usability of a CA.

2.9.2.1 Evaluating CA Usability and Quality of Experience (subjective metrics)

Many metrics with regards to satisfaction from the users' perspective can be determined via a questionnaire (Kitchenham and Pfleeger, 2002). According to (Albert and Tullis, 2013) a user experience metric reveals something about the interaction between the user and the product, some aspects of effectiveness (being able to complete a task), efficiency (the amount of effort required to complete a task), or satisfaction (the degree to which the user was happy with his or her experience while performing the task). A questionnaire can be utilised to highlight the user's opinion on the following subjective attributes:

- Usability (Laugwitz et al., 2008)
  - Ease of use
  - Interface design
  - Language used
- Performance/User satisfaction
  - Was the goal/task achieved (Pietquin and Hastie, 2013)
  - Was the information helpful
  - Was the information given understandable
  - Time taken to reach aim/goal
  - Was the CA approachable/intuitive
  - CA naturalness (Lutfi et al., 2013)
  - Domain knowledge coverage

Questionnaires are a commonly used tool for the user-driven assessment of software quality and usability. They allow an efficient quantitative measurement of product features (Laugwitz et al., 2008). A commonly deployed questionnaire design to measure user satisfaction with relation to software quality and user experience is the Likert scale style questionnaires (Laugwitz et al., 2008, Hassenzahl, 2008, Hassenzahl et al., 2010). This type of questionnaire is easy for the users to understand, more importantly it is quick to complete (Lee et al., 2014, AlSanad, 2014). Participants involved in end user evaluations are administered a questionnaire subsequent to their interaction with the software in order or gauge their perceptions to the metrics measured through the questionnaire questions, either through written responses or through their level of agreement with a particular scale related to a certain metric (Laugwitz et al., 2008).

2.9.3 Objective Evaluation Methodologies

According to O'Shea et al. (2011) most research includes a set of objective measures which are used to test research hypothesises and goals. Generally speaking, there is a leap of faith that these in some way reflect the aspirational subjective measures that appear at the beginning of published studies. The only systematic and scientific approach was that taken by the PARADISE framework (Walker et al., 1997). All recent work makes use of some of the fundamental PARADISE measures whilst adding some application-specific elements which will be the approach adopted in this evaluation.

2.9.3.1 Evaluating CA Quality of Service (objective metrics)

Objective metrics can be measured through records and logs of the user's dialogue with the CA. These metrics are captured whilst a user is undergoing an evaluation session to achieve a pre-set task. The records/logs are used to capture and store several variables related to the dialogue such as rule fired, similarity strength, user utterance, CA response etc. By utilising this information the following attributes can be measured and analysed:

- Dialogue / Conversation length and path complexity (O'Shea et al., 2011)
- Conversation success and goal achievement (Hassenzahl, 2008)
- Effectiveness of the algorithms
    - ability to reduce the number of scripted patterns and scripting time
    - effectiveness to calculate similarity strength
    - utterance recognition accuracy measures

Based on these captured variables which are stored in the log file, the CA can be evaluated for effectiveness accuracy and robustness, through statistical analysis.

2.10 Conclusion

This chapter has introduced the concept of conversational agents as software applications that facilitate communication between a user and a computer system using natural language. Historical developments within the field have been described and conversational agents have been reviewed in terms of functionality. The earlier CA's such as ELIZA were not what is considered to be intelligent, they were general chatbot's, designed with the sole aim of continuing the conversation with the user, without a goal or aim to the discussion. The ALICE chatbot relies on a large knowledge base of rules for general conversations, but for goal-based situations such as tutoring, InfoChat is more powerful and the features of the PatternScript language offer more sophisticated scripting of a CA. The more recent implementations of CA's such as InfoChat focused on adding some intelligence to the agent in order for the agent to be able to conduct dialog with the user in order to reach a goal.

The many challenges that are inherent in developing CA's have been outlined, such as the labour-intensive and time-consuming development and maintenance of CA scripts that are one of the layers that make up the CA knowledge base. The methodologies and processes involved in creating CA a knowledge base have been described. Finally, traditional CA evaluation methodologies have been review and possible alternative approaches have been described and will be considered in evaluating the new proposed Urdu CA.

The next chapter will provide an in depth overview of the Urdu language. The grammatical and morphologic complexities are discussed, as well as the unique challenges that are inherent in implementing Urdu in a CA.

Chapter 3 - Urdu Language

3.1 Introduction

Urdu is the national language of Pakistan, home to about 180 million people. Globally, it is spoken by over 60 million people in more than 20 countries including Pakistan. Urdu, an Indo- European language of the Indo Aryan family, is spoken in India and Pakistan.

Among all the languages in the world it is most closely similar to Hindi. In the same way that Hindi has adopted many words from Sansikrit the classical version of Hindi, Urdu has borrowed a large number of vocabulary items from Persian (Farsi) and Arabic (Hardie, 2003). Arabic and Farsi languages have close resemblance with Urdu, but Urdu is more complex compared to Arabic and Farsi due to additional characters (Khan et al., 2012).

Limited Urdu language support exists. There are currently no Urdu chatbots or CA's. The Urdu language is grammatically and morphologically much more complex when compared with English and other western languages, therefore in order to develop an effective CA it is important to examine and outline the complexities of the language and how these complexities affect the implementation of an Urdu CA.

3.2 Written Style of Urdu

Urdu is written in Arabic script with some additional characters which are not present in Arabic language (Durrani and Hussain, 2010). It is a bidirectional language. Sentences start from the right side and numbers are written from left to right (Abandah et al., 2014). This bidirectional nature of language increases the complexity of Urdu writing system. While spoken, Urdu is quite similar to Hindi but is absolutely different in writing. Whereas it is written form is more similar to Persian and Pashto. Urdu differs from Arabic in writing because it uses more complicated and convoluted Nastaliq script, mostly used for Urdu orthography and Arabic leads to follow a more modern nashk/script. Nastaliq is actually written from top right to bottom left (Abdul-Mageed and Korayem, 2010). As a consequence the Urdu script is more difficult to read than Arabic, and introduces more complexity in parsing written Urdu, as this feature makes it more difficult to distinguish word boundaries. This is illustrated in Figure 6 where the same passage of text is written in both written styles.

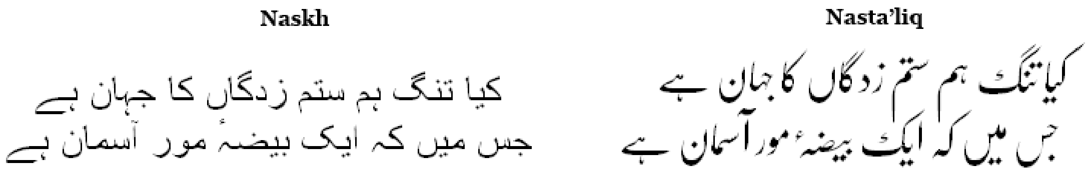


Figure 6 - Urdu text written in both styles

Urdu alphabet is comprised of 38 basic characters illustrated in Figure 7. These characters are joined together to make words of the language. Nastaliq is most widely used and is defined by well-formed rules passed down through generations of calligraphers. Nastaliq was originally created by the calligrapher Mir Ah Tabrezi, and has been refined by master calligraphers over the past 600 years. Nastaliq is derived from two other styles of Arabic script Naskh and Taliq (Iqbal et al., 2011). It was therefore named Naskh-Taliq, which is shortened to Nastaliq.

Figure 7 - Urdu alphabet (38 letters) and numbers

3.3 Written System

The most obvious visual qualities of the Urdu script which it shares with Arabic is that it is written horizontally from right to left, and that it is always cursive, even in its printed form. As a consequence characters are realised differently depending on their position in the word (initial, medial, or final), and the majority of characters are joined to the characters on either side in writing and in print. There are also some ligatures that have special forms which occur when particular characters appear together in a word. Urdu has an alphabet of 38 basic letters most of which have dots known as “nokhtas” above or below them and 15 diacritical marks known as “areab” or “harakat” (Naz et al., 2014a).

3.4 Nastaliq Writing Style

Two most common feature of Nastaliq found in Nashk or for that matter in any Persian or Arabic script is that it is cursive. Another characteristic is that Nastaliq is written from right to left unlike English which is from left to right (Naz et al., 2014a). In addition to these, there are other characteristics of Nastaliq that have made its automation, printing, computational analysis/processing difficult because the Nastaliq style is written from top left and flowing down to the bottom right, this method of writing proved difficult to implement and standardise in modern computing due to its difficult writing style. This has led to the adoption of the standard nashk that is used in Arabic and Persian computing.

The modern nashk/script writing style was adopted as it is easier to read and better suited for computational use. The nashk writing system is written from right to left, however it does not follow the Nastaliq writing styles that flows from top right to bottom left, nashk is written on a straight line. The more modern nashk made it easier for Urdu to be implemented computationally. As there is a Unicode character set available for the modern nashk which includes the symbols and diacritics that are unique to Urdu. This means it should not pose a problem to implement Urdu and all the new characters and diacritics within a conversational agent as a full Unicode character set is available.

3.5 Diacritics

The Arabic, Persian and Urdu languages have a large set of diacritical marks that are necessary for the correct articulation of a word (Farukh and Vulchanova, 2014). The diacritical marks appear above or below a character (illustrated in Figure 8) to define a vowel or to geminate a character (Malik, 2005, Zia, 1999). They are the foundation of the vowel system in these scripts (Malik et al., 2010). The diacritics in Urdu represent vowels sounds, stops and pauses. Figure 7 below illustrates how the diacritical marks are used in conjunction with Urdu consonants.

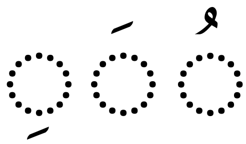


Figure 8 - Position of diacritical marks around consanants

Figure 9 illustrates the consonant letter ب “Bey” which is equivalent to the English letter “B”. In each of the forms illustrated in Figure 9 the letter sound changes due to the addition of diacritical marks.

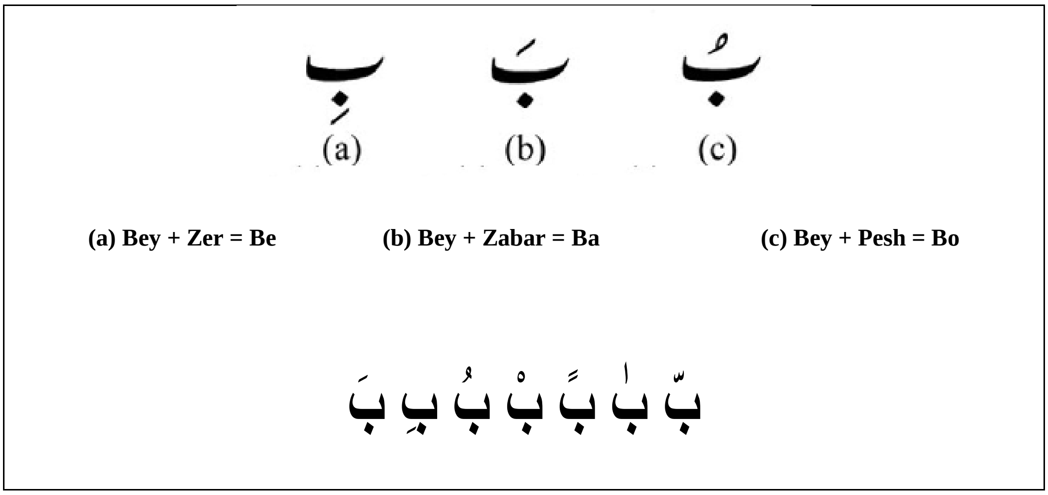


Figure 9 - Urdu diacritical marks with the consonant bey

Figure 9 outlines how the different vowel sounds change the pronunciation of the consonants depending on the diacritical mark that appears above or below the consonant. The diacritical marks have the same effect on each of the consonants.

In Urdu script, the consonantal context is clearly represented, but the vocalic sounds are represented (mostly) by diacritics. The consonants are written but the vowels are not always written explicitly, for example the word '*diacritics*' would be written as '*dcrtcs'*, and both '*ball'* and '*bill'* will be written as "*bl"*. The vowels are realized through the diacritical marks above or below the preceding consonant but are optional and normally not written. Native speakers can normally recreate these unwritten vowels through contextual knowledge (Raza and Hussain, 2010) and thus can pronounce words correctly, based on their knowledge of the language. But un-diacritized Urdu text creates ambiguity for novice learners and computational systems (Raza and Hussain, 2010).

As the number of vowels in Urdu is considerably greater than that of Arabic, the three marks Fatha, Kasra and Damma (in Urdu referred to as Zabar, Zer and Pesh respectively) are insufficient. Urdu uses these marks to represent the three short vowels and a combination of these marks with base characters ا، و، ی ے to indicate the long vowels (Whaley, 1996). Urdu uses additional diacritical marks which are illustrated in Figure 10.

بّ بٰ بً بْ بُ بِ بَ

Figure 10 - Additional diacritical marks used in Urdu

Figure 11 illustrates how an example passage of text looks both with and without the associated diacritical marks.

Text without diacritics

پاکستان کے شمالی علاقے سربلند چوٹیوں سرسبز وشاداب وادیوں پہاڑوں کو چیرتی آبشاروں رومانی جھیلوں

دیو قامت گلیشیرز بل کھاتے دریاؤں اور گھنے جنگلوں جیسے قدرتی حسن سے مالا مال ہیں۔

Text with diacritics

پاکِستَان کے شُمَالی عِلاَقے سَربُلَند چوٹِیوں سَرسَبز وشادَاب وَادِیوں پہاڑوں کو چِیرتی آبشَاروں رُومَانی جھِیلوں

دیو قامَت گلیشِیرز بَل کھَاتے دَریَاؤں اَور گھَنے جَنگلوں جَیسے قُدرَتِی حُسن سے مَالاَ مَال ہَیں۔

Figure 11 - Same text written with and without diacritics

All of the Urdu diacritical marks, though part of the writing system, are sparingly used (Zia, 1999). They are essential for disambiguation, natural language processing and speech synthesis (Malik, 2005, Malik, 2006, Malik et al., 2008). For Native speaker and readers of the language diacritical marks are seldom used, however for people who are new to reading Urdu the diacritical marks make reading and pronunciation easier (Jawaid and Ahmed, 2009).

3.6 Ambiguity

The Urdu language has ambiguities just as English where one word can have more than one meaning (e.g. bank financial institution or the side of a river). In addition to this type of lexical ambiguity, Urdu, due to the nature of its script has another feature which introduces ambiguity. Further lexical ambiguity arises due to the absence of diacritical marks in written Urdu. The diacritical marks represent vowel and stops/pauses as discussed in section 3.5.

3.7 Word Order

One of the noteworthy aspects of Urdu grammar constitution is its word order SOV (subject, object, and verb). This order does exhibit some flexibility as the subject pronouns are frequently dropped (Hardie, 2003, Naim, 1999). The basic word order of the Urdu (SOV) is an extremely common word order in the world’s languages (Whaley, 1997). However, word order in Urdu is relatively free (Butt et al., 1994b), variation in word order is common, particularly the reordering of small elements for thematic purposes (Kachru, 1990).

It should be noted, that Butt (1995) among others has argued that Urdu is non-configurational, that is, the ordering of elements of the sentence is not restricted (Naim, 1999). Bögel and Butt (2013), provide further substance to this notion, they state that Urdu is a free word order language, meaning major constituents of a sentence can reorder freely. A single sentence in Urdu can be expressed in multiple ways and still be grammatically correct. This notion is also shared by Raza (2011), who states Urdu is a free word order language. The verb in a sentence usually (but not always) comes last and its arguments are put in any order before it. An example of this is illustrated in Table 1 where variation 2 is almost always used but the others are legitimate.

English Sentence	**I (subject)**	**am**	**angry**	**at**	**Raheem (object)**
Urdu 1	mujhe (subject)	gussa	dilate	hai	raheem (object)
Urdu 2	raheem (subject)	mujhe (object)	gussa	dilate	hai
Urdu 3	raheem (subject)	gussa	dilate	hai	mujhe (object)
Urdu 4	gussa	dilate	hai	mujhe (subject)	raheem (object)
Urdu 5	raheem (subject)	mujhe (object)	dilate	hai	gussa

Table 1- Demonstrating free word order in the Urdu language

This loose/free word order carries on through to either asking a question or giving a reply in Urdu, the same question can be asked in many different ways and still sound grammatically correct within conversation.

The case markers in Urdu mark the grammatical roles or functions to the words, with which they are attached. Usually, they are lexically attached through inflection or derivation. But, in Urdu language, the case markers are independent lexical units and are treated as independent parts of speech (Rizvi and Hussain, 2005). They influence the sentence structure and can cause grammatical ambiguities, For example, the free word order property of the Urdu text is due to lexically independent case markers, e.g., both the phrases; "رنگوں کے نام" (rangoon kay naam, colours' names) and "رنگوں کے نام" (naam rangoon kay, names colours') are accurate with the same meanings, but have different word order because of the case marker "کے" (kay equivalent to "of" in english) (Abdul-Mageed and Korayem, 2010).

This variance word order is a significant issue in a pattern matching conversational agent. This is because the user utterance is matched to a database of previously compiled responses. In a language where there is no strict word order, it means that the domain will have to be scripted to compensate for all the different possible responses and variation in word order. This will result in extensive script writing which will be a lengthy and time consuming task.

3.8 Word segmentation

In computation the process of splitting and dividing a sentence/string of characters into individual words, is technically known as word segmentation or tokenization (Mahar et al., 2012). The tokenisation of words is the preliminary step in any system of natural language processing, where the initial phase requires tokenization of input into individual words (Durrani and Hussain, 2010, Rashid and Latif, 2012). Once the words are segmented, different applications and processes can be developed. Almost every application of NLP requires, at certain stages, the process of breaking its text into individual tokens for processing -for example, in Machine Translation (MT) and Spell Checking (Haruechaiyasak et al., 2008). The tokenization process is done by identifying word boundaries in languages like English where punctuation marks or white spaces are used to segregate words. In Languages such as English, French, Hindi, Napali, Bengali, Greek, Russian etc. space, commas and semi colons can be utilised for word boundary identification. However, many Asian languages like Thai, Khmer, Lao, Dzongkha and Urdu do not have strict word boundaries and thus do not use white space to consistently mark word endings (Durrani and Hussain, 2010).

Urdu and its sister Asian languages like Arabic and Persian, endure the same problem of text segmentation with space omission and insertion issues (Bhatti et al., 2014). Hence, white space is not a concrete indicator to identify word boundaries, making the segmentation/tokenisation of Urdu strings challenging. One has to use high level information such as semantics and word morphology of the language for word segmentation (Rashid and Latif, 2012). The concept of word spacing in Urdu is explained by Durrani (2007) who states; “the notion of space between words is completely alien in Urdu hand-writing”. Children are never taught to leave space when starting a new word. These orthographic issues are resolved by humans using natural intelligence applied to contextual information. They just implicitly use the rules and the human lexicon to know when to join and when to separate. It has been established that space is not a reliable tool for marking the word boundary for Urdu text, this is due to the morphological nature of the script. In Urdu script, unlike English script, space is not used to separate two consecutive words in a sentence; instead readers are able to distinguish the boundaries of words as they read along the text (Akram and Hussain, 2010).

Urdu script is based on the Arabic script, which is cursive, meaning the letters in the script join together into units to form words. These connected units are called ligatures. The cursive nature of Urdu text is illustrated in Figure 12. The figure shows just how different the isolated characters are when compared to the written cursive form.

Isolated spelling	Cursively written
ب ا د ش ہ ی م س ج د	بادشہی مسجد

Figure 12 - Isloated and cursively written versions of sample text in Urdu.

Translates to: "Badshahi Masjid (Kings Mosque)" the name of a mosque in Lahore Pakistan.

The very interesting phenomenon that Urdu and Arabic script exhibits is change in basic shapes of characters. This change of shape is dependent on the position of the character in ligatures. This leads to four possible positional categories in which shapes of a character can be divided. These are initial, medial and final positions of a character in a ligature and the isolated one. One character can acquire several shapes in each position. The shapes of a character are dependent on characters coming before and after it (Butt et al., 1994a). Urdu characters change their shapes depending upon neighbouring context. But generally they acquire one of the following four shapes:

1. isolated
2. initial
3. medial
4. final

Furthermore, Urdu/Arabic alphabet characters can be divided into two groups, joiners and non-joiners (Naz et al., 2014b). The joiners can be connected cursively within a word ligature for example the word Kaleem in Urdu is spelt entirely of joiners as illustrated in Figure 13. It can be seen from the figure that the whole word is written cursively without any breaks in the ligature as it contains all joiner characters.

كليم

Figure 13 – Cursive ligature with all joiner characters (Kaleem)

However the non-joiners can only be connected from the right hand side, and proceeding characters start from the initial position, as illustrated in Figure 14. It can be seen from the figure that the ligature is broken where non-joiner character is used with the word.

بادشاه

Figure 14 – Cursive ligature with non- joiner characters (King)

Each letter has multiple forms depending on its position in the word. Each letter is drawn in an isolated form when it is written alone, and is drawn in up to three other forms when it is written connected to other letters in word.

For example, letter ‘Khah’ has four forms: isolated (1), initial at the beginning of a word (2), medial in the middle of a word (3) and final at the end of a word (4) illustrated in Figure 15.

خ	خـ	ـخـ	ـخ
Isolated	Initial	Medial	Final

Figure 15 - Four forms of the letter khah

The non-joiners character of the Urdu alphabet can acquire only isolated and final shape illustrated in Figure 16.

ے و ژ ز ڑ ر ذ ڈ د ا

Figure 16 - Non-Joiners in Urdu

On the other hand joiners can acquire all the four shapes. The isolated form of each of these is shown in Figure 17.

ض ص ش س خ ح چ ج ث ٹ ت پ ب

ہ ی ن م ل گ ک ق ف غ ع ظ ط

Figure 17 - Joiners in Urdu

The Urdu Nastaliq style of writing does not have the concept of space to separate words. Similar to South-East Asian scripts like Lao, Thai and Khmer, Urdu readers are expected to parse the ligatures into words as they read along the text. In typing, space is used to get the right character shapes. To achieve this end, it is sometimes used within a word to break the word into constituent ligatures (Akram and Hussain, 2010). However, if the ligature form is achieved without the use of space (i.e. the preceding ligature ends in a non-joiner), it is sometimes not even used in between two words if it is a visually correct sequence of two words for the reader.

For example when writing sentence **"بادشاہی مسجد کا دروازہ بند ہے"** (The door of Badshahi Mosque is closed). A native speaker knows that the word '**بادشاہی**' ends in a joiner and the following word '**مسجد**' begins in a joiner so the writer would start a new word by inserting a space (similar to English). In this case if no space is used the ligatures will merge into one, which does not look or read correctly.

With space	Without space
بادشاہی مسجد	بادشاہیمسجد

Figure 18 - Example of text when word ends in joiner

Figure 18 illustrates how when a word ends in a joiner and the next word starts in a joiner space must be inserted otherwise the two words merge and look incorrect. However, the word '**مسجد**' ends in a non-joiner character "**د**" thus the writer has the option not insert a space as the following ligature would not be affected by the former words ligature as illustrated in Figure 19. The two words would still look and read correctly.

With space	Without space
مسجد کا	مسجدکا

Figure 19 - Example of text when word ends in non-joiner

In Urdu typing, space is only used to get appropriate character shapes and sometimes it is even used within a word to break the word into constituent ligatures. Therefore, for Urdu language processing, word segmentation or word tokenization is preliminary task for understanding meanings of the sentences (Akram and Hussain, 2010). Currently there are no automatic word segmentation utilities available for Urdu (Hussain and Durrani, 2008) and little to no other computational resources (Sinha and Hyma, 2013).

3.9 Common Spelling Mistakes/Variation

A study of English by Damerau (1964) states that 80% of the typographic errors fall into one of the following four categories:

1. Single letter insertion; e.g. typing acress for cress
2. Single letter deletion, e.g. typing acress for actress
3. Single letter substitution, e.g. typing acress for across
4. Transposition of two adjacent letters, e.g. typing acress for caress

Two studies have been carried out by Naseem and Hussain (2007) , to identify common spelling error patterns in Urdu. Their study concluded that in Urdu, spelling errors exhibit a couple of script specific trends that are not found in the studies of error trends of English. One of these is the frequent occurrence of substitution errors caused due to the shape and phonetic similarity of the letters in Urdu alphabet. The other is the omission of spaces at word boundaries. They also state that their results will apply to other languages that are written in Arabic script and that their results imply that the existing rule based spelling correction algorithms may not be as effective for Urdu, and for Arabic script based languages in general, as they are for roman script languages. The existing techniques require modifications or redevelopment to cater the script specific issues of Urdu spelling errors (Naseem and Hussain, 2007).

3.10 Lack of resources

There have been many factors causing slow growth of Urdu software. One of the contributing factors has been the lack of standards for Urdu computing such as a standardised Unicode character set (Hussain and Afzal, 2001). Ahmed and Butt (2011) argue that one of the major bottlenecks for development is the lack of lexical resources available for the Urdu language. For example the Urdu language doesn't have the established electronic infrastructures that is taken for granted in English and other European languages, such as lexicons, annotated electronic dictionaries, corpora and well-developed ontologies that describe relationships among words and entities in written text (Naseem and Hussain, 2007).

The development of linguistic Conversational Agent's (CA's) has primarily been focused on the English and other European Languages. There is limited existing research for the Urdu language and no Urdu Conversational Agent's (UCA) exist. This lack of resources has a major impact on the implementation of an UCA as it narrows down the development options available to implement certain architecture components such as Sentence Similarity Measures (SSM).

3.11 Conclusion

It can be concluded from the existing work and literature review conducted in this and the previous chapter that the research into CA's has been focused on mainly English and western languages (Alobaidi et al., 2013). Research into other languages is still in its early stages and other languages do not have the extensive lexical infrastructures that are required to implement some CA components (e.g. STS, WordNets). PM remains the predominant methodology for scripting in poor resource languages, as other CA development methodologies require sophisticated components which in turn require computational resources which are still not readily available in non-western languages.

It has been established that Urdu has certain distinctive features, like; variable vocabulary and grammatical rules, independent case marking and context sensitive script (Abdul-Mageed and Korayem, 2010) as well as word segmentation issues that will pose as challenges when implementing the language in to a CA. In addition to complex morphology, the Urdu language has some other distinctive features, which make it a challenging language to implement into a CA, e.g., influences from various languages, lexicon intricacy, context sensitivity of the script, and free word order due to independent case marking (Abdul-Mageed and Korayem, 2010).

Furthermore, greater challenges arise due to Urdu being a computationally resource poor language. From a historical perspective research in to NLP and CA scripting has been performed for the most part in English, including the design of scripting engines, scripting methodologies, resources and implementation procedures. An example of this is that, existing English engines rely on word spacing to differentiate between words; however this is not always the case with other languages e.g. in Urdu and Arabic. In addition to this English has weak use of gender compared with Urdu, as the pronunciation of nouns in Urdu language grammar have two types of gender (masculine/feminine) depending on whether the sentence is referring to a man or a woman (Anwar et al., 2006).

It is evident that the word order rules in the Urdu language poses some novel challenges to overcome when implementing Urdu in a conversational agent. In Urdu there are many ways to say the same thing using the same words in a different order, as discussed earlier. One possible method of overcoming this issue could be, parsing the user utterance and arranging the utterance into a standard format and then matching it to responses from a database of possible responses. The response can be matched exactly as the user has input it, and a parsed version can be matched and the best match of the two will be used to trigger an appropriate response.

This variance word order is an important issue in a pattern matching conversational agent. This is because the user utterance is matched to a database of pre-empted responses. In a language where there is no strict word order, it means that the domain will have to be scripted to compensate for all the different possible responses and variation in word order. This will result in extensive script writing which further exacerbates an already lengthy and time consuming task.

Urdu script is cursive and context sensitive. Urdu alphabets are categorized as joiners and non-joiners. In a word, the joiner alphabets join with each other in different shapes according to their position in the word. If the ending alphabet of a word is a joiner then it tends to connect with the first letter of the next word, resulting into a misidentification of the word boundaries (Abdul-Mageed and Korayem, 2010). Hence, this context sensitivity results in word segmentation issues (Lehal, 2010), as the spaces are not always exact indicators of the word boundaries, as in case of English. In written Urdu space does not necessarily mean a word boundary, however in computation most users have accepted the limitation of technology in this case and accepted space as a separating character. In other case where the user does not want the space to be visible uses zero-width non-joiner character (U+200C; ZWNJ). Nevertheless this makes the problem a little relaxed because with this character the text contains some clues in form of space or by using the non-joiner character encoding (ZWNJ) about where a potential word boundary is (Durrani, 2007). Where a space occurs within a word the non-joiner character is used, which can be used to differentiate between word boundaries and non-joiners. The Urdu conversational agent engine should be able to differentiate between the non-joiner character, and a word boundary to ensure the engine is able to pattern match accurately.

Urdu diacritical marks can help with disambiguation of certain words, however the need to implement diacritics is not an essential feature. The CA is aimed at people with a firm grasp of the Urdu Language, who are computer literate, thus they will be well versed in using and communicating with Urdu without the need of diacritics. However some words in the Urdu language are homographs without diacritics, this issue will have to be addressed when the domain is scripted. The domain will have to be thoroughly researched and all possible cases of homographs will need to be handled by the CA engine based on the context of the discussion.

Unfortunately, morphologically rich languages (MRLs), i.e., Arabic, Turkish, Urdu, Finnish, etc., are relatively overlooked by the research community, because in these languages, the word level complexity is very high due to the frequent morphological operations (Abdul-Mageed and Korayem, 2010). The existing English language CA engine’s such as ALICE and InfoChat do not have to deal with language features such as free word order, inconsistent word spacing/segmentation, diacritical marks and common spelling variations. It is for these reasons that existing conversational agent engines, scripting methodologies, algorithms and approaches, developed for other well-studied languages, are not effective for Urdu text. Therefore, it is not feasible to simply take an existing engine designed for English and adapt it for Urdu. As Syed et al. (2014) state "the morphological complexity and flexibility in grammatical rules of this language require an improved or altogether different approaches in NLP application development". A whole new architecture is needed to support conversation in the Urdu language. Hence, it requires different approaches to engine design, scripting and algorithms for the implementation of the Urdu language in a Conversational Agent, which can incorporate the issues highlighted efficiently to produce a functional Conversational Agent.

In light of the issues highlighted, a new methodology and algorithms are required in order to develop a conversational agent in the Urdu language, which can handle the language specific issues of this morphologically rich and resource poor language (Mukund et al., 2010, Khan and Buchanan, 2014) with the intention of delivering an effective and coherent discussion. The problem of scripting being a laborious task will be exacerbated when implementing a CA in Urdu, as the research has highlighted there is an issue of free word order which means one sentence has several legal permutations, in addition to the inconsistent word segmentation issue. This means that the scripting could grow exponentially depending on the size of the selected domain. Furthermore, other language unique features such as diacritical marks and ambiguities pose additional challenges in order to correctly parse and process the user utterances to overcome. In addition to this Urdu also poses different challenges to the development of an Urdu CA, lexical computational resources are scarce at best, meaning some of the more recent developments in CA design such as WordNets cannot be utilised. Moreover, another fact about Urdu to take into consideration is that the language has no capital letters for proper names: the names of people, countries, cities and names of months or days of the weeks like English. This increases the inability to detect key words and classify them.

The following section will outline and detail the proposed architecture for creating the new UMAIR Urdu CA architecture.

Chapter 4 - Developing a Conversational Agent for the Urdu Language

4.1 Introduction

Based on the research conducted in to the development of CA’s and the complexities and language unique challenges posed by the Urdu in Chapter 2 and 3 respectively, it has been established that there are limited Urdu language processing resources available in order to implement an Urdu CA and to the researchers knowledge, to date no Urdu chatbot’s or CA’s exist.

This chapter outlines and details the framework and architecture components of the proposed Urdu Conversational Agent (UCA) called UMAIR (Urdu Machine for Artificially Intelligent Recourse). UMAIR is a novel Goal Orientated Urdu Conversational Agent developed to mimic a customer service representative for the National Database and Registration Authority (NADRA) of Pakistan. UMAIR will offer users advice and instructions on ID card application related issues. The first phase of this research aims to develop a prototype, proof of concept UCA that demonstrates novel CA components their ability to alleviate the challenges of the Urdu language.

A novel Goal Orientated Conversational Agent framework designed specifically for the Urdu language will be developed, the framework is comprised of a novel scripting language, string similarity algorithm and CA architecture. The architecture will encompass novel components to deal with the language unique challenges of Urdu (detailed in Chapter 3) such as the Word Order Wizard (WOW) algorithm.

4.2 UMAIR CA Framework Overview

The UMAIR framework is made up of a novel Urdu scripting language and CA architecture. The Urdu scripting language implemented within the framework is based on the principles set by the InfoChat CA scripting language (see section 2.6). However as the InfoChat scripting language was designed for use with English CA’s, further developments and enhancements to the scripting language have been made in order to create a scripting language suitable for the Urdu language. The main features of the Urdu scripting language are:

- The scripting language provides Urdu dialogue for UMAIR
- Link rules and patterns to answers and supporting material
- Hold rule specific variables such as allowed rules and next rule to fired that allow the scripting language to work with the knowledge trees
- Provide links to the knowledge tree nodes
- Ability to provide supporting material to the user
- Ability to control the conversation through context
- Ability to switch the conversation context based on certain predefined rules firing
- Ability to elicit further responses from the user to extract further information
- Ability to script a conversation path to ensure the user is lead towards the goal of the conversation
- Ability to allow WOW or not depending on the pattern content and context

The key components and features that comprise the architecture of UMAIR are as follows:

4.2.1 Novel Urdu Engine

- Responsible for pattern matching and calculating the similarity strength between the user utterance and the scripted patterns (section 4.5.2).
- The engine will utilise a hybrid approach which utilises lexical sentence similarity measure (WOW) and pattern matching techniques in order to calculate the matching strength between the user utterances and the scripted patterns.

4.2.1.1 Conversation Manager

- A state machine that is responsible for controlling and directing the conversation through contexts which represent different stages throughout the discussion.
- The conversation manager is also responsible for ensuring the discussion is always directed towards the goal of the discussion to make sure the goal of each discussion is met.

- Ensuring that the knowledge tree has enough information from the user utterance to make its traversal towards the goal (leaf). If enough information is not received the conversation manger asks the user additional questions to obtain the necessary information required.

4.2.1.2 Utterance Cleanser

- The filter is responsible for normalising the user utterance before it is processed by the engine. The normalisation process involves removing diacritics and other illegal characters (e.g. ! £ $).
- The filter also ensures that only appropriate language is used by the user by inspecting the user input for offensive words/key words and takes appropriate action in the event that unacceptable language is used, such as warning the user or ending the conversation/discussion.

4.2.2 Knowledge Base

- The knowledge base is responsible for holding all the domain knowledge in a relational database which includes:
  - Scripts, rules and patterns separated into contexts.
  - Knowledge trees based on business logic of the domain.
  - Supporting material
    - images
    - sounds
    - documents
  - Urdu language specific knowledge such as offensive words, interrogative, agreement and disagreement words, in order to provide the CA with some semantic information.

4.2.3 Graphical User Interface

- Facilitate communication between the users and the agent through a chat like interface, with additional interface panels to display supporting material.

The framework outlined in this section is illustrated in Figure 20 which also illustrates how these components work together in UMAIR's engine.

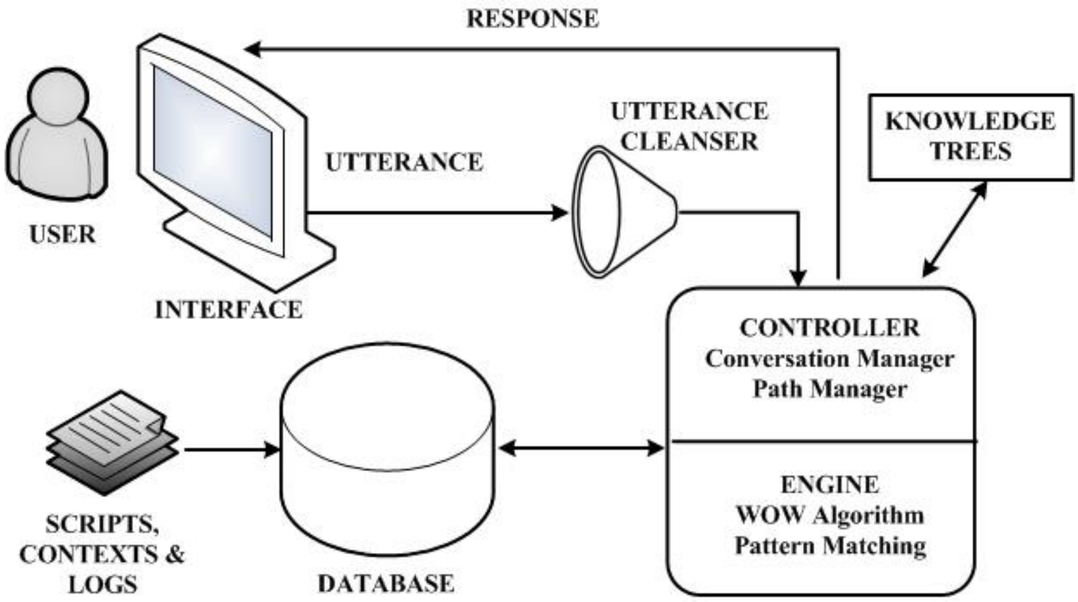


Figure 20 – Proposed UCA architecture diagram

4.3 Methodology for Implementing the UCA

The UCA was designed and implemented according to the GO-CA development methodology proposed by O'Shea et al. (2011) this software development methodology combines elements of the staged approach used in the Waterfall model with elements of prototyping or iterative development. The major stages are shown in figure 3 below.

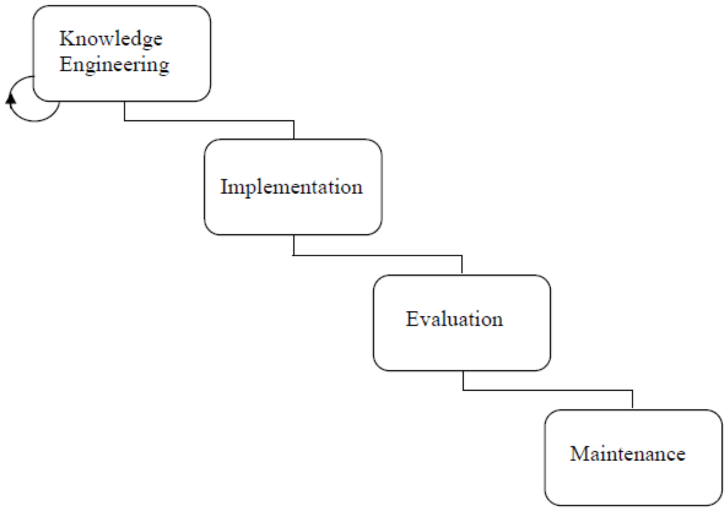


Figure 21 - GO-CA Software Development Methodology

The development of the UCA will follow 4 phases; which are the following:

- Phase 1: Creating and structuring the Urdu Scripting Language and Knowledge Base (section 4.4)
    - Design knowledge base for UMAIR
    - Conversation design
    - Scripting the domain
- Phase 2: Implement the UCA framework (section 4.5)
    - Develop UCA engine
        - PM
        - Similarity Algorithm
    - Controller
    - Conversation and Path Manger
    - Offensive Language Inspector
    - Utterance Cleanser
    - Temporal memory
    - GUI
- Phase 3: Implement UMAIR (section 4.6)
    - Conduct knowledge engineering for the selected domain
    - Construct knowledge trees
    - Script conversation in knowledge base
- Phase 4: Testing and Evaluation of the UCA (section 4.7)
    - Subjective end user evaluation (questionnaire)
    - Objective end user evaluation (Log files)
    - Data analysis

These four phases will be followed in order to incrementally research and develop each component of the UCA architecture and finally evaluate the effectiveness of the architecture as a goal orientated Urdu conversational agent. Phases one and two are focused on creating a domain independent Urdu CA architecture. Phases three and four are focused specifically on implementing and evaluating a domain specific CA (UMAIR) by utilising the framework and architecture developed in phase one and two. The developments choices and resulting components from each phase are described in detail in the following sections.

4.4 Phase 1: Creating Urdu Scripting Language and Knowledge Base

The first phase of the research involved creating a knowledge base for the UCA. For a CA the knowledge base is considered to be the brain of the system. The knowledge base consists of knowledge related to the domain which needs to be organized in an understandable fashion. This knowledge was then scripted and stored within the CA system to act as the backbone to the conversation between the user and the system.

4.4.1 Urdu Scripting language

The foundations of UMAIR's scripting language is based on the Info Chat (Michie and Sammut, 2001) scripting language. The InfoChat framework included a complex formulae to calculate the matching strength between scripted patterns and user utterances. The formula utilised several variables such as which space, activation level and number of words to determine the match strength. The InfoChat engine expected consistently segmented words with white space, and did not have to deal with diacritics. However, the approach adopted by InfoChat did not consider semantic or word/string similarity, thus making this approach obsolete compared to newer conversational agent engines.

In the UMAIR PM engine the similarity strength is calculated through the novel WOW similarity algorithm (see section 4.5.3) which combines the use of the Levenshtein edit distance algorithm to compute the similarity strength between words in the utterance and the scripted pattern while providing novel features which can handle unique features of the Urdu language such as the free word order which is solved using the Bipartie Graph and Khun Munkers methods (see section 4.5.2). The pattern with the highest matching strength will fire its corresponding rule and the controller relays the answer/response back to the user.

The scripting language includes a feature that allows it to provide supporting material to the user. Depending on the need of the user the scripting language allows supporting material to be conveyed to the user in the form of images, application forms etc. This material is stored in the scripting database and once a rule is fired, if that rule has material to support the user's query it is delivered to them through the interface. This can be a map image showing their local passport office which is displayed on screen or a document related to their query such as an application form that they can then download. This adds another dimension of support to the UCA and makes it seem more helpful and intelligent to the user, as opposed to just providing responses strictly in text form.

The scripting language works with the controller to check whether the fired rule requires the context of the conversation to be changed. This is handled by the Switch Context variable in the scripting language. This is a variable that is stored in the database, any fired rule that has this variable associated with it, switches the context of the conversation to a different sub topic (an example of UCA scripting language is illustrated in table 3) where the Switch to variable is the tree node that the conversation topic is then switched.

Context General – Application Form	Context General – Application Form
Rule – App_Form Pattern: * form do I need for new ID card Pattern: which form * for ID card Pattern: I need a form * ID card Pattern: * form for new ID card Response: The form to apply for an ID card is the POC form. You can either download a form, or visit your local NADRA office where you can pick one up. Switch Context: null Switch to: null Support material: poc_form.pdf Requires Vars: No Allow Yes/No Tree Node: null Allow WOW: Yes	Rule – App_Form Pattern: کی تشکیل میں نئے شناختی کارڈ کے * ہے لئے کی ضرورت Pattern: شناختی کارڈ کے * لیے جو فارم Pattern: مجھے ایک فارم کی ضرورت ہے * شناختی کارڈ Pattern: نئے شناختی کارڈ کے لئے * فارم Response: ایک شناختی کارڈ کے لئے شکل ہے. آپ کو یا POC درخواست دینے فارم تو ایک فارم ڈاؤن لوڈ، یا آپ کو ایک ہی اٹھا سکتے ہیں جہاں آپ کا مقامی نادرا کے دفتر کا دورہ کر سکتے ہیں. Switch Context: null Switch to: null Support material: poc_form.pdf Requires Vars: No Allow Yes/No Tree Node: null Allow WOW: Yes

Table 2 - Extract of UCA scripting language

Furthermore, there are certain questions asked by UMAIR that can be answered with a simple yes or no from the user within the system. However in some instances a yes/no answer is not sufficient enough for the system to be able to make a firm knowledge tree traversal decision (see section 4.6.3 knowledge tree). An example on this would be when UMAIR asks the user which documents they may have to prove their citizenship. In this instance the user could just say “yes”, to indicate that they have a certain document. However which particular document they have dictates UMAIR’s response, accordingly the scripting language includes a novel feature called the AllowOneWord rule, which tells UMAIR that a detailed answer is required for that question in particular. If the user simply answers yes/no then a linking question is delivered back to the user in order to elicit more information from them with regards to the current context.

The scripting language also works with the knowledge trees implemented in the system. The scripter is able to direct the flow of the conversation by relating each scripted rule to a node of the knowledge using the Tree Node feature of the scripting language. This feature is included in the scripting language to allow the engine to know which state the conversation is in and which paths the conversation can follow in order to reach the goal/leaf node of the current context. This allows the scripter to structure the conversation in the knowledge engineering phase by analysing the domain to construct knowledge trees and subsequently utilise them to aid with the scripting phase. This method of implementing the conversation allows the scripter to predetermine and control the conversation flow from conversation initialisation to the conversation goal. The controller uses the tree node feature of the scripting language to determine the path the conversation must follow in order to reach the conversation goal. The path is loaded when the initialisation rule of a certain context is fired. The controller then checks each subsequent rule that is fired to ensure it is following the correct conversation path. This feature of the scripting language allows the knowledge tree logic to be embedded in to the scripts, which can then be processed by the Conversation Manager (section 4.5.8 for conversation manager).

Another feature implemented in to the Urdu scripting language is the ‘allow wow’ rule. It has been established through the research that Urdu is a relatively free word order language (section 3.7). However just as with English, if certain words in sentences are moved to another part of the sentence, the meaning of the sentence is changed. Consequently, the allow wow feature enables the scripter to disable the WOW algorithm processing of that rule during run time. This will reduce incorrect matching with patterns that cannot be classed as free word order patterns through the WOW algorithm.

4.4.2 Scripting Methodology

In order for a domain to be implemented within the UCA, a new Urdu scripting language had to be developed. In section 2.6, numerous approaches to CA development have been identified and discussed. Due to several key impediments (i.e. lack of available resources and unique characteristics of the Urdu language) it was concluded that PM was the most appropriate approach to develop UMAIR.

A number of approaches to CA development and inherent challenges that come with designing a CA for the Urdu language were discussed in section 3.11, through this research it was concluded that pattern matching is the most suited for the development of UMAIR. The domain was scripted using the knowledge trees created in the knowledge engineering phase as a guide to the questions and possible dialogue that could occur during that stage of the conversation. Each node in the decision tree maps to at least one rule in the script database and has multiple scripts/patterns that could invoke that particular node/rule as illustrated in Figure 22.

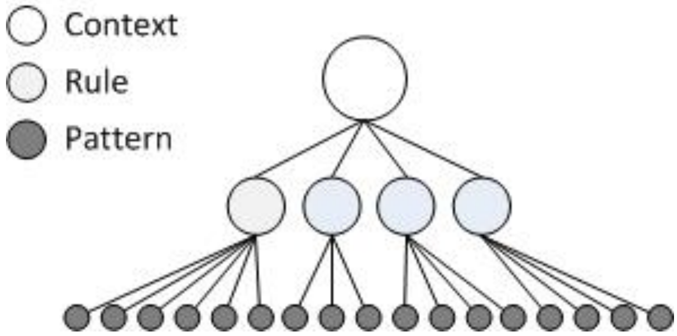


Figure 22 – Patterns mapped to rules

Conversational agents require scripting for particular domains, which is a time-consuming and complex task (Latham et al., 2010b). The UCA domain is scripted using the new scripting language, which is stored within a relation database in order for the system to retrieve and utilise them. The new scripting language and the new UCA engine are designed to deal with the challenges and complexities of the Urdu language.

To represent the domain within the UCA, a new Urdu Scripting Language had to be developed. This new scripting language took initial ideas from the InfoChat scripting language (Michie and Sammut, 2001) which was designed to allow simple scripts to be developed in the English language.

The domain was structured into contexts and each context contained rules, each created rule in the domain contains a number of patterns that match the user utterance to the patterns stored within the database, and a response that forms the CA output/response to that utterance.

In addition, the scripting language contains features that allow it to work with the conversation manager to allow for context switching and supporting material to be displayed to the user, an example of one scripted pattern is illustrated in Figure 23. The features of the new scripting language are explained in section 4.4.1.

Context 1 - ID Card
Rule 1 – Need new ID Pattern: I * new ID Pattern: I need a new ID because * Pattern: I need a * ID Pattern: How * new ID Response: Are you a citizen of Pakistan? Switch Context: Yes Switch to: Sub context 1.1 New ID Card Support material: No Requires Vars: No AllowYN: No TreeNode: 2

Figure 23 - Extract of UCA scripting language (translated)

The procedure used to create the scripts within the database followed an adapted/customised approach based on the scripting methodology devised by Latham (2011).

The procedure followed for scripting each context was as follows:

- Create a new record in contexts table with a unique name to represent that context.
- Create an initialisation rule that fires when the context is invoked.
- Script all rules and patterns for the associated context.

- Test the individual context to check that rules fire when expected, and amend any patterns as necessary to avoid conflict with other patterns in the same context.

The procedure followed for scripting each rule was as follows:

- Create a unique rule name and create a new record in rule table for the expected user utterance based on each new node in the knowledge tree.
- Consider the user utterance. Extract the important words and create a pattern to match the utterance, using the wildcards to replace unimportant words.
- Consider all possible ways of phrasing the utterance, e.g. saying the same things using alternative words create patterns for each different phrase.
- Script the CA response/answer to the utterance.
- Add context switching parameters the rules which when fired allowed the context of the conversation to be changed
- Add additional appropriate helpful/accompanying resources to the rule, such as images and documents which are displayed on screen to aid the conversation.

The above procedure was utilised to script the knowledge for the whole domain, it allowed the scripting to follow a systematic and structured process by following the knowledge trees. However due to the WOW algorithm one major step proposed by Latham (2011) which was "Consider different ways of phrasing the utterance, e.g. using words in a different order, and create patterns for each different phrase", could be removed from the scripting methodology as the WOW algorithm dealt with this during run time.

In order to implement this knowledge in to the UCA architecture, a database structure was designed. The database was implemented using MySQL, the database schema is illustrated on Figure 24.

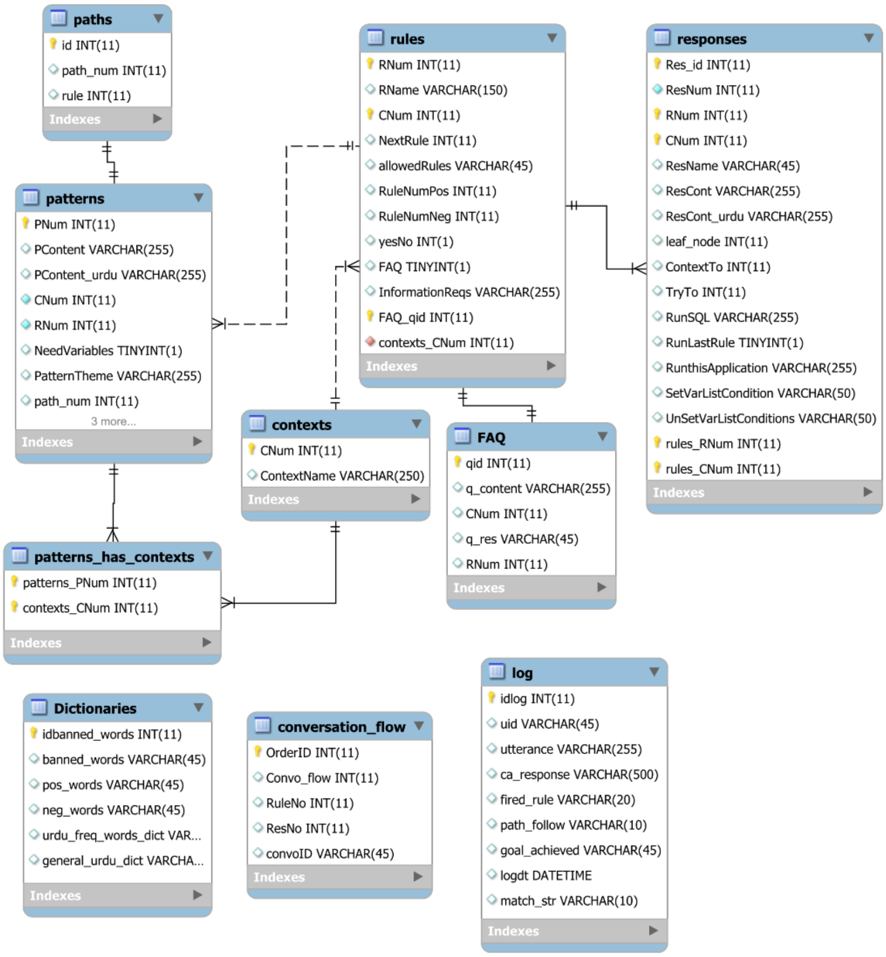


Figure 24 – Knowledge base database schema

The knowledge base was implemented in a relational database, which allows the scripting of all the dialogue for the agent and other knowledge base resources such as the FAQ layers and dictionaries to be stored and accessed by the engine.

The previous sections outlined the scripting language and how it was used to script the domain specific knowledge within the knowledge base which is considered to be the brain for the UMAIR through which the CA is able to converse with the user.

4.5 Phase 2: Implement the UCA Framework

The UCA architecture illustrated in Figure 20 was implemented. The prototype UCA components were developed using Microsoft C# and ASP.Net programming languages. The knowledge base and script databases were developed using the ConvAgent tree tool and MySQL. The functions of the individual components of the UCA architecture are described in detail in the following sections.

4.5.1 Urdu CA Engine Components

The primary aim at this stage of the research is to develop a novel CA architecture, design specifically for the Urdu language. A General overview of the architecture and how its components interact to process a user utterance in Urdu is illustrated in Figure 25.

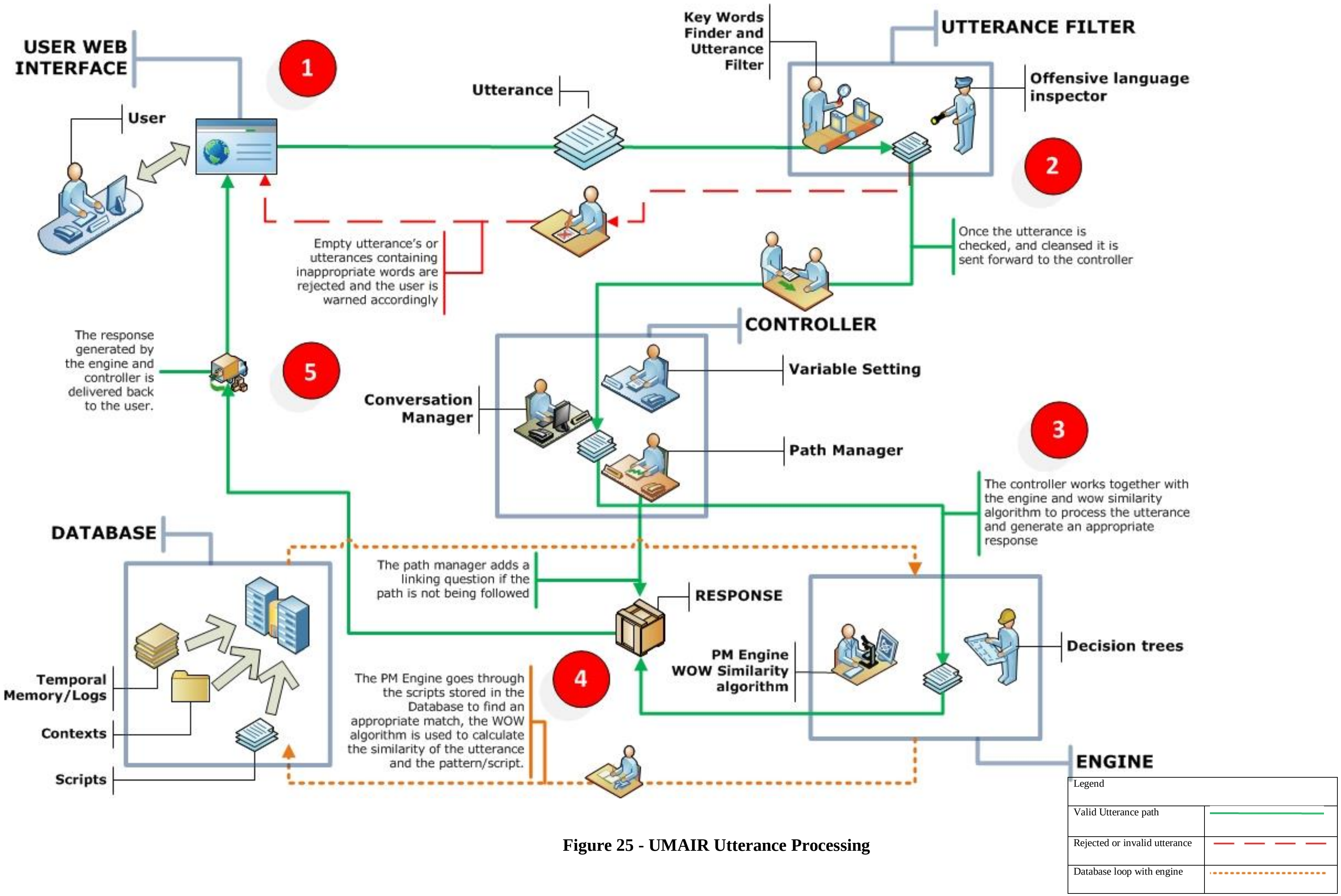


Figure 25 - UMAIR Utterance Processing

4.5.2 Pattern Matching Engine Components

UMAIR introduces a novel method to determining the similarity between two sets of strings within CA's, while traditional CA's utilises a PM based approach that involves strength calculation through different aspects of the user utterance and the scripted pattern such as activation level and number of words etc. The UMAIR UCA utilises string similarity metrics to overcome some of the challenges in the Urdu language. The primary phase of the engine is based on similar concepts to the Info Chat (Michie and Sammut, 2001) method of PM where the user utterance is matched to stored scripts which contain wild card characters to represent any number of words of characters. The second phase utilises the novel WOW sentence similarity algorithm which considers the lexical similarity of the individual words as well as the variation in word order, in order to calculate a similarity strength between the user utterance and stored scripts.

From the background/literature research it was found that one of the most prominent challenges that came with implementing the Urdu language in a CA that needed to be addressed was the issue of free word order. As discussed in section 3.7 of the Urdu language review the Urdu language has a relatively loose/free word order. This poses a big challenge because the UCA implements the PM approach for CA development as there is a distinct lack of resources available for the Urdu language (Ahmed and Butt, 2011). The PM approach requires precompiled scripts that define the conversation to be executed by a pattern-matching engine. The scripts contain rules which in turn contain patterns (O'Shea et al., 2011).

It is a well-known fact within the field of CA development that scripting is the most laborious and time consuming part of CA development (O'Shea et al., 2008). The biggest challenge of scripting CAs is the coverage of all possible user utterances (Latham, 2011). This challenge grows exponentially when a CA is implemented in the Urdu language as the free word order means one utterance can be said many different ways.

This is a significant language specific issue; it would make scripting a CA in Urdu a much more laborious task which would take significantly longer than scripting in a language with a fixed word order such as English. The new engine architecture comprises of components that work together to analyse the user utterance and provide the appropriate response. These components include a Wild Card PM Function combined with a novel WOW similarity algorithm that calculates similarity strength and handles the word order problem. The WOW similarity algorithm was designed to satisfy the following requirements which make it suitable to be used in the UCA:

- It should be robust enough to handle changes in word order - two strings which contain the same words, but in a different order, should be recognized as being similar.
- Consideration of lexical similarity - strings with minor differences should be recognized as being similar. In particular, a significant sub-string overlap should point to a degree of similarity between the strings (i.e. user utterance and scripted pattern).

The UCA engine was designed based upon a number of features that were specifically developed to deal with the features unique to the Urdu language in terms of its morphological nature and grammatical nature. The UCA engine developed incorporates a number of novel features, which can be described as follows:

- Pattern matching function (Wild card PM).
- Novel WOW string similarity algorithm which comprises of:
    - Levenshtein Algorithm (word similarity) (section 4.5.2.1).
    - Bipartite Matching (word order variance) (section 4.5.2.2).
    - The Kuhn-Munkres algorithm (also known as the Hungarian method or the “matching problem”), used to find the maximum match strength between two sets of strings (section 4.5.2.3).

The combination of these two components within UMAIR’s engine come together to form a novel CA PM engine that calculates the similarity of the user utterance with scripted patterns using string similarity metrics in addition to taking word order into consideration. Therefore reducing the need to cover all possible word order variations when scripting the domain.

4.5.2.1 Levenshtein Algorithm

In many applications, it is necessary to determine the similarity of two strings. A widely-used notion of string similarity is the edit distance: the minimum number of insertions, deletions, and substitutions required to transform one string into the other (Ristad and Yianilos, 1998). Levenshtein Distance (LD) is a popular algorithm to compare strings by various edit operations devised by Vladimir Levenshtein (1966), usually including the deletion, insertion, and substitution of individual symbols (Sankoff and Kruskal, 1983). This measure is often called the "edit distance" and can be defined as the minimum cost of transforming one string into another through a sequence of weighted edit operations (Li and Liu, 2007). Transformations are the one-step operations of insertion, deletion and substitution. If the source and target are identical the cost is zero. A single insertion or deletion to one string to make it match costs one unit and substitutions cost two units.

$$lev_{a,b}\,(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j) = 0, \\ \min \begin{cases} lev_{a,b}\,(i-1,j) + 1 \\ lev_{a,b}\,(i,j-1) + 1 \\ lev_{a,b}\,(i-1,j-1) + [a_i \neq b_j] \end{cases} & otherwise. \end{cases}$$

Equation 1 - Levenshtein edit distance algorithm

4.5.2.2 Bipartite Matching

The matching or assignment problems are one the fundamental classes of combinatorial optimization problems. In its most general form, a matching or assignment problem can be stated as follows: a number of agent's *n* and a number *m* of tasks are given, possibly with some restrictions on which agents can perform each particular task. A cost is incurred for each agent performing some task, and the goal is to perform all tasks in such a way that the total cost of the assignment is minimized (Dasgupta et al., 2008).

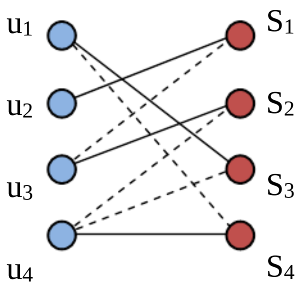
$$G = (U, S, E)$$

Equation 2 - Bipartie Graph Algorithm

Equation 2 denotes a bipartite graph whose partition has the parts U and S, with E denoting the edges of the graph. The partition of the two string in to a bipartite graph allows the two sides of the graph to be compared to one another for similarity. The similarity strength between the two sides of the graph are then used as the edge weights, which are utilised in the next step by the Kuhn-Munkres algorithm to find the maximum match strength between the two sides of the bipartite graph.

4.5.2.3 Kuhn-Munkres algorithm

The Kuhn-Munkres algorithm, also known as the assignment problem, is a widely-studied problem applicable to many domains (Burkard and Cela, 1999). The Kuhn-Munkres algorithm assumes the existence of a bipartite graph, G = (U, S, and E) as illustrated in Figure 26 where U and S are the sets of nodes in each partition of the graph, and E is the set of edges. The edge weights are stored in a matrix as shown in Figure 27. Missing edges are assigned to have zero weight (Mills-Tettey et al., 2007). Assuming that numerical scores/weights *E* are available for the similarity of each of *U* tokens on each of *S* tokens, the "assignment problem" in this instance is the quest of finding the largest score so that the sum of the scores so obtained is as large as possible (Kuhn, 1955). The larger the total scores the stronger the similarity between the two sets of tokens/words or utterance and scripted pattern.



w_{ij}	u_1	u_2	u_3	u_4
s_1	0	1	0.4	0
s_2	0	0	1	0.3
s_3	1	0	0	0.6
s_4	0.5	0	0	1

Figure 26 - A bipartite graph

Figure 27 – Matrix of edge weight

Matches between tokens means that there could be many possible ways to match, or link tokens from a database pattern to a user utterance. To match each database pattern to at most one user utterance token/word, the items in the sentence pairs are modelled as nodes in a bipartite graph and use the Kuhn-Munkres algorithm (Munkres, 1957, Kuhn, 1955) to find a maximum weight matching (or alignment) between the tokenised words in polynomial time. The weights ((w_{ij}) from the edges) of the resulting graph will then be added to determine the final similarity score between the pair of sentences, which is output as a floating point value between 0 and 1.

The Kuhn-Munkres algorithm transforms the problem of word order from an optimization problem of finding a max-weight matching into a combinatorial one of finding a perfect matching. It combines the edge weights assigned to find the maximum matching strength. This is a classic technique in combinatorial optimization.

4.5.3 The Word Order Web (WOW) Similarity Algorithm Overview

The WOW algorithm calculates similarity in three steps by utilising the algorithms described in the previous section:

1. Partition each string into a list of tokens providing a bipartite graph. Tokens are separated by whitespace characters firstly and then each token is validated with the Urdu dictionary. Any unrecognised token is processed by the word segmentation algorithm (see chapter 7 section 7.4, word segmentation algorithm).

Given a graph G (U, P), G can be partitioned into two sets of disjoint nodes U (left tokens/utterance) and P (right tokens/pattern) such that every edge connects a node in U with a node in P, and each edge has a non-negative weight (Secer et al., 2011) which is determined by the edit distance (see section 4.5.2.1). U is the set of the first list of tokens from the cleaned and normalised user utterance. P is the set of the second list of tokens from the cleaned and normalised database patterns. E is a set of edges connecting between each couple of nodes/vertices (U, P), the weight of each edge which connects an u_1 to a p_1 is computed by the similarity of u_1 token and p_1 (see example below).

$$\begin{aligned} user\ utternce&:\ u_1 u_2, \dots u_n \\ database\ pattern&:\ p_1, p_2, \dots p_n \end{aligned}$$

2. Computing the similarity between tokens by using a string edit-distance (Levenshtein) gives the each token its edge weight. This computes the similarity of the words in the two token lists.
3. Computing the similarity between the words in the two token lists. This is to address the variations in word order. This is handled by the bipartite graph algorithm; the maximum weight is calculated using the Kuhn-Munkres

algorithm which is then returned as a float value between 0 and 1. The maximum weight denotes the final similarity score between the two strings.

4.5.4 WOW Algorithm Explanation/Walkthrough

After the user utterance and pattern have been split in to two separate token lists, the first similarity check uses the Levenshtein edit-distance string matching algorithm. The string edit distance is the total cost of transforming one string into another using a set of edit rules, each of which has an associated cost. The similarity method checks similarity the between two token lists (i.e. user utterance and pattern from the database). After splitting each string into token lists, the similarity between two sets of tokens is computed. This is reduced to the bipartite graph matching problem.

$$w(i,j) = Lev\ (token\ [u^n], token[p^n])$$

Equation 3 - Algorithm for computing weights of tokens/node edges

The calculation returns a score which is between 0 and 1. The closer the score is to 1 the higher the similarity, which means that if the score gets a maximum value (equal to 1) then the two tokens/words are identical. This score is then utilised as the edge weight.

The final task is to find a subset of node-disjoint edges that has the maximum total weight, the higher the total weight the closer the similarity of the two strings being compared. The similarity of two strings is computed by the number of matched strings in both token lists in the bipartite graph. The results of this function are used to compute the weight (*w*) of edges which are then initialised and stored within a matrix of edge weights (illustrated in Figure 27).

The edges are then connected to find maximised the total weight, which is then divided by the number of words in the utterance to give the final similarity score which is a floating point number between 0 and 1. This step is handled by the Kuhn-Munkres algorithm which is used to find the maximum total maximum weight of bipartite matching which is divided by the number of words in the utterance (u_{max}) to return a float value between 0 and 1. The closer the value is to 1 the stronger the match between the two compared strings.

$$\max weight = \frac{\sum \max_{0 \le i \le j} (w_{i_n, j_n})}{u_{max}}$$

Equation 4 - Kuhn-Munkres maximum weight of edges of biparte graph

A maximal weighted bipartite match is found for the bipartite graph constructed, using the Kuhn-Munkres Algorithm, the intuition behind this being that every keyword in a sentence/utterance matches injectively to a unique keyword in the other sentence/pattern, if it does not then the highest match weight is utilised as that token/nodes edge weight. Thus, the final similarity strength score (*sim*) between sentences user utterance (U) and pattern (P) is:

$$sim\ (u, p) = \frac{Max\ (tokens\ (u), tokens(p))/2}{Maximum\ Sum\ of\ Edge\ Weights}$$

Equation 5 – Strength Similarity Algorithm

A high level overview of this process is illustrated in the Figure 7 where the WOW algorithm is applied to an example user utterance and the process of the similarity strength calculation is illustrated.

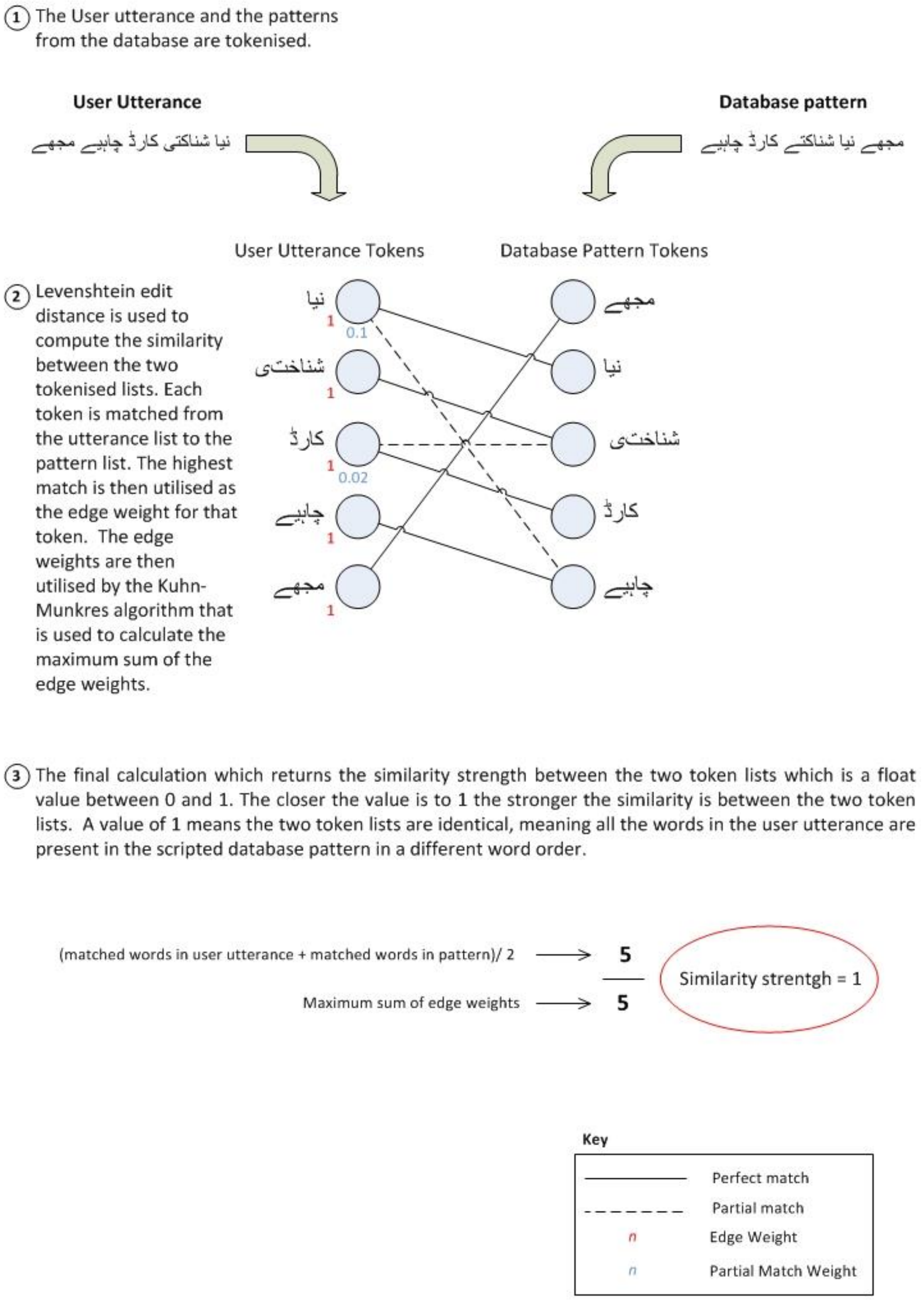


Figure 28 - WOW algorithm walkthrough with example

4.5.5 Significance of the WOW Algorithm

The combination of the algorithms explained in the preceding sections, solves the complex word order issue that comes with the Urdu language. It also significantly reduces the number of scripts that have to be scripted to deal with the issue of word order an example of this is illustrated in Table 3.

Scripted pattern	* Mujhe مجھے	neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے
Patterns covered	* Mujhe مجھے	shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے
	* Mujhe مجھے	shankthi card شناختی کارڈ	chahiye چاہیے	neya نیا
	*Neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے	mujhe مجھے
	* Shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے	mujhe مجھے
	* Mujhe مجھے	Chahiye چاہیے	neya نیا	shankthi card شناختی کارڈ

Table 3 - Word order variation in a single scripted pattern (translation: I need a new ID card)

The issue of word order is a major obstacle when it comes to implementing the Urdu language within a PM CA. Table 3 illustrates how a single utterance can be expressed in many different ways in Urdu. This was a major challenge for the UCA to overcome as this matter makes it very difficult for the scripter to script the domain as all possible word order variations have to be pre-anticipated.

Moreover, this will drastically increase the time it takes to script the domain, which is seen primarily as one of the major drawbacks to implementing PM CA's. Through the implementation of the new WOW similarity algorithm the UCA is able to overcome these challenges and PM all the word order variations on a single scripted pattern in the database, hence saving the scripter major time and effort. The researcher is well aware that word order variance can change the meaning of the intended utterance, however to control such ambiguity the UCA implements techniques to control the conversation through contexts. The UCA is aware of the current context of the discussion, which helps overcome misunderstandings in word order as well as ambiguity through synonyms.

4.5.6 Utterance Sentiment Classifier

The Sentence Sentiment Classifier is a feature in engine that allows UMAIR to classify each of the user utterances into either a positive or negative utterance. In some cases it is important to get the sentiment of the user utterance in order for the engine to be able to utilise the decision trees. The decision trees are utilised by conversation and path manager in order to lead the conversation towards a predefined goal (see section 4.6.3) each node in the decision tree represents a state/context of the discussion. The goal is achieved once the decision tree reaches a leaf node. The sentiment classification feature allows the UCA to be able to use the decision tree rules to give the user the relevant responses to their particular query based on the sentiment they express in their utterances.

The utterance is classified in the following steps:

- The utterance is parsed and tokenised into its individual words
- The tokenised words are then compared to two word tables in the database one containing positive words and the other containing negative words
- Each match is tallied and calculated to produce a totally weight value to classify the whole utterance
- The highest total after calculation determines the utterance sentiment (positive or negative) if they are equal or no matched words are found then they utterance is classified as neutral in which case depending on the previous rule fired if a classification is absolutely necessary for UMAIR to continue the discussion the pervious question is repeated and the user is instructed to use different words because UMAIR was not unable to understand.

$$Classification = (+n\ words\ )(-\ n\ words)$$

Equation 6 - Utterance classification equation

4.5.7 The Controller

The controller is responsible for directing and managing the entire conversation. The controller is the core of the CA and works in conjunction with several other components to ensure the conversation goal is achieved. The controller is also responsible for delivering an intelligent, cohesive and goal led conversation. Before the utterance is passed to the PM engine the controller is responsible for processing user utterances based on the following parameters:

- Check for Bad/Rude/Inappropriate words, and warn the user or terminate session depending on how many times unacceptable language is used in the session (See section 4.5.10).
- Respond to empty input by asking the user to interact using the textbox on screen. If the user continues to pass empty utterances the session is terminated.
- Cleansing the utterance. The controller uses the utterance filter to remove special characters (i.e. $, &, *, !, ?, “”, £, (), ^) from the utterance (See Utterance Filter section 4.5.11).
- Making sure the input language is Urdu. If the user enters anything other than Urdu the controller instructs the user to either use the on screen keyboard (see section 4.5.12 on screen keyboard) if they do not have the Urdu keyboard installed on their particular system or to switch their input language to the Microsoft Urdu keyboard.

After the utterance is parsed the controller then works together with the conversation and path manager (see section 4.5.8) to ensure the conversation is following the correct path, or whether the context needs switching. Once this is complete the controller is responsible for delivering responses back to the user and where necessary any accompanying supporting material such as pictures or documents according to the fired rule.

4.5.8 Conversation and Path Manager

The role of the Conversation Manager (CM) is to control the flow of the conversation to ensure that the goal is achieved. Depending on the context the CM loads a predefined path stored in the database that ensures the goal of each context within the domain is met during the conversation. The conversation manager ensures that the user stays on topic, and manages the switching of the contexts during the discussion by working together with the Path Manager (PM) component. The PM loads a path stored in the database which is a predefined path with the aim of reaching a desired goal within the context of the conversation. The path is defined through the knowledge tree nodes feature of scripting language which the PM reads when the initialisation rule for that context fires. For example if the user states that they have lost their ID card, the *lost_id* path will load in to the memory of the path manager. The goal of the path is to lead the user through the conversation and give them all the information they require to be able to know how to replace their lost id card.

Another aspect handled by the PM is the ability to handle utterances that are not related to the current context of conversation. Goal-oriented CAs must employ mechanisms to manage unexpected utterances in a way that appears intelligent (Latham, 2011). If the path manager receives an utterance that is not in the path of the current context, the path manager checks the user utterance with the FAQ knowledge layer then checks to see if the utterance matches other contexts within the database. If a match is found the utterance is responded to, and then the user is brought back to the point where the conversation digressed and directed towards the goal again.

In addition to this the path manager is able to handle instances where the user asks a question that is addressed later in the conversation. For example if the conversation is in the context of the 'how to acquire a new ID card', the expected path is as follows:

1. Are you a citizen of Pakistan?
2. Have you ever had an ID card before?
3. Do you have any of the following supporting documents?
4. Have you filled in the application form?

If for example the user is asked question 1 but instead of answering this question the user instead asks "which supporting documents are required for application?", the path manager is able to recognise that this is related to question/rule number four and it is expected to come later on in the discussion. Thus, the UCA will answer the question then remove question/rule number four from the path, and then bring the user back to question one with a linking question, thus regaining control of the discussion and directing the user accordingly (Illustrated in Figure 29). The removal of the rule from the conversation path stops the UCA repeating itself and makes it seem more intelligent to the user by not asking the same question twice or repeating something that has already been covered. This is an important feature for a CA to have as it was found in the literature that repetition from a CA lowered end user satisfaction (Silvervarg and Jönsson, 2011, Walker et al., 1997). It was also found that in some instances such as CA's used for tutoring systems repetition was a positive feature, however in a helpdesk/customer service environment where the conversation length is much shorter than that of a tutoring session, repetition is perceived as unintelligent and makes the CA seem less intelligent. In light of this, the path manager is able to dynamically adjust the conversation path based on the user utterances and rules fired.

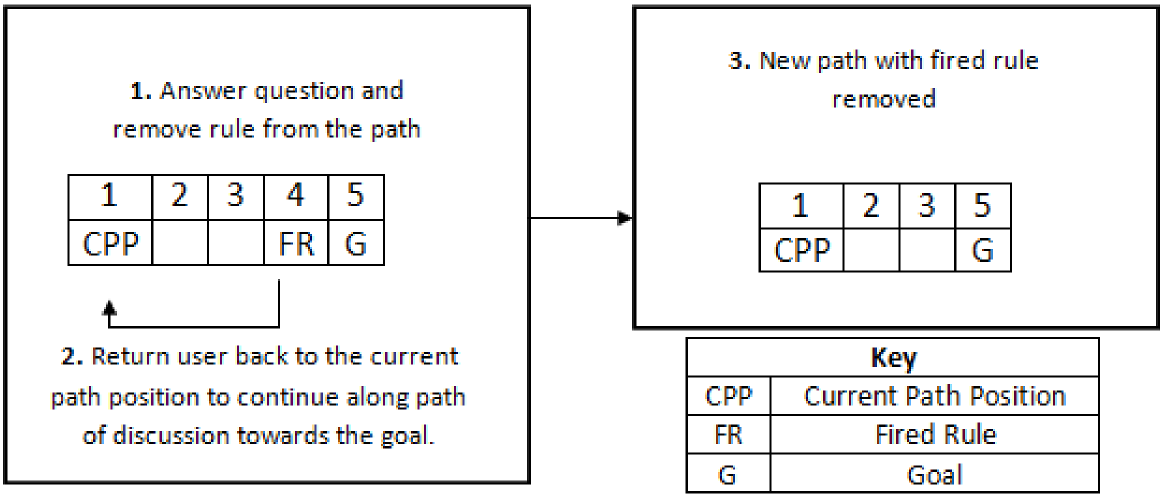


Figure 29 - Path Manager managing rules in conversation path

4.5.9 Temporal Memory (Log File)

The UCA will utilise a temporal memory/log file feature, which will allow it to store several variables and conversation related information in a database table. The information captured and stored in the temporal memory database can be utilised to evaluate the system and track end user conversations. The following information will be recorded in the temporal memory database.

- User utterance
- CA generated response
- Rules fired during the conversation
- Similarity strength
- WOW algorithm induced utterances
- Bad words/utterances
- Number of utterances not recognised by the CA
- Positive or negative utterance classification
- Conversation goal achieved.

4.5.10 Offensive Language Inspector

The Offensive Language Inspector is able to recognise and respond to bad/inappropriate language used by the user. The knowledge base database has a table in which all inappropriate words are listed; the controller validates every word of the user utterance with this list to ensure the utterance does not contain any bad language. The list was formulated through the interview with the industry contact. During the interview the industry contact provided insights in to how the NADRA customer services representatives are trained to deal with unacceptable/inappropriate behaviour and language, and what is deemed to be unacceptable language.

When the UCA finds an inappropriate word within the utterance, the CA responds with a warning to the user to refrain from using bad language and that the system does not tolerate abusive language or behaviour. If the user persists to use unacceptable language after the first warning the session is terminated, and the GUI is disabled to stop further discourse.

4.5.11 Utterance Filter

The utterance filter is responsible for normalising the user utterance by removing special characters (i.e. $, &, *, !, ?, “”, £, (), ^) from the user input such as diacritics and punctuation (see section 3.5 for detailed explanation of diacritics in the Urdu language). One of the features of the Urdu language is the use of diacritics which represent the vocalic sounds when applied to the consonant characters. However the use of diacrital marks is entirely optional as native speakers are able understand the words without diacritical marks through the contextual information. This feature of the language creates additional challenges for an Urdu PM CA as the scripts will have to account for the text with and without the diacritical mark. Therefore to reduce the scripting effort the utterance filter removes all the diacritical marks from the user utterance before it is sent forward to the engine for processing. The filtered utterance is then sent to the PM engine to process. The filtering ensures that only clean and consistent input is sent forward for pattern matching. This also makes scripting the domain easier as the scripter does not have to anticipate punctuation and or other diacritical marks which can be entered by the user.

4.5.12 Onscreen Urdu Keyboard

During the early stages of the research it was found that the Urdu language was not as established as English and other languages in terms of computation (Hussain and Afzal, 2001) its only recently the Urdu has been standardised in Unicode and a Urdu keyboard layout has been included in Microsoft Windows. In terms of Urdu on the web, it could only be achieved by using specialised software such as Urdu InPage to write the Urdu and then converting the typed document into an image file which will then be displayed on the website (Khan et al., 2012). Due to this the number of computer users who actually have the standard Urdu keyboard installed and activated on their PC's is very limited. Thus, in order to overcome this problem, UMAIR has a custom built on-screen keyboard, which contains all the characters of the Urdu alphabet (illustrated in

Figure 30).

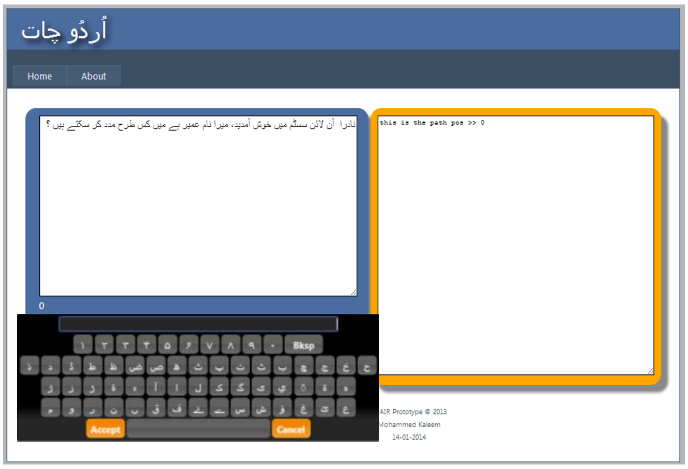


Figure 30 - UMAIR Custom On-screen Urdu Keyboard

If the user opts to use UMAIR's on-screen keyboard, it will result in a further advantage from a PM perspective, which is the input is restricted to only legal characters thus further reducing the chance of an utterance containing something that will lower the effectiveness of the similarity score. Furthermore this makes UMAIR accessible to more people as the users can communicate, in their native language easily as keyboards that have the correct letters displayed on each key are seldom found.

4.5.13 Graphical User Interface (GUI)

The GUI is the point where the UMAIR and the user interact with each other. The GUI takes in user input/utterances from a textbox and delivers the generated responses back on to the interface. The UCA GUI is also able to display images/maps and deliver supporting material such as electronic documents and forms to the user. This makes the conversation more stimulating and provides the user with necessary material related to their query, making the CA more helpful and relevant to the user's situation.

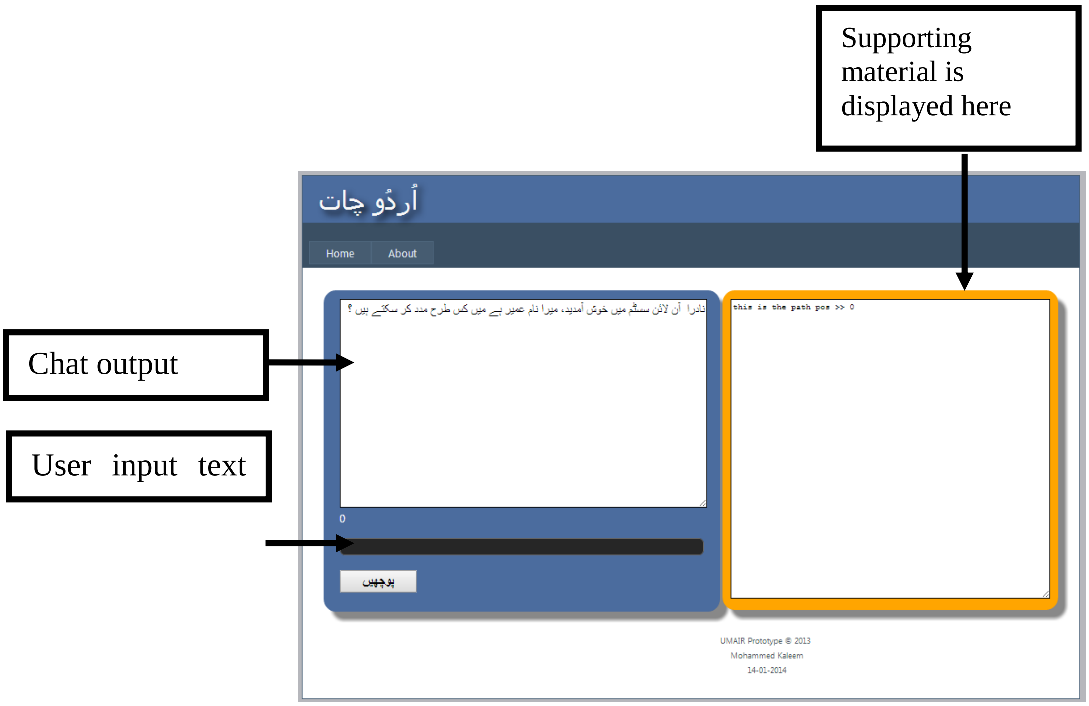


Figure 31 - UMAIR Main Interface

4.6 Phase 3: Implementation of UMAIR

Utilising the Urdu Conversational Agent framework outlined in the previous sections the UMAIR was implemented. The framework created is a domain independent framework designed to handle the language unique challenges of Urdu. In order to test the framework a domain was selected in to which UMAIR was deployed as a customer service representative.

4.6.1 The Domain

The National Database and Registration Authority (NADRA) ID card application is the domain selected for the prototype UCA (UMAIR). NADRA is a government run organisation in Pakistan that deals with the registration of all of Pakistan's citizens. NADRA is the responsible for registering Pakistani citizens and overseas nationals in to its central database. Once individuals have registered with NADRA they are able to apply for ID cards, passports and other identification related documents such as birth and family certificates. NADRA receives a very high volume of customer queries on a daily basis, therefore a conversational agent in a customer service role will be very beneficial to customer relations and customer information access.

After the main research of the domain and knowledge engineering, the knowledge was used to construct the main questions in relation to ID card application asked by the customers/users of this service. Thereafter, a rule base was then used to structure the ID card context within the knowledge base.

4.6.2 Knowledge Engineering the Domain

The knowledge base for the UCA was extracted, designed and developed based on existing business logic used within the selected domain's organisation. In order to get a good grasp of the domain a short interview was conducted with the industry contact in NADRA Pakistan to gain some first-hand insight into the domain and the frequently arising issues in the selected problem domain.

Subsequent to this, the domain was further investigated through available sources on the internet and through a second more in-depth interview with an employee (Mr Kashif Iqbal, Lead IT Manager) working for the NADRA ID card application department in Islamabad, Pakistan. During the second interview the researcher was able to ask the NADRA representative about typical customer related queries and scenarios that the customer service representatives have to deal with throughout a working day. The information from this interview was transcribed and utilised in creating conversation structure through contexts and formed the basis of the evaluation scenarios used to test the system (see Appendix J for interview questions).

The information from the research and interviews was then collated and converted to process flow charts (see Appendix D) which were sent to the industry contact in Pakistan for verification and approval to ensure the domain/business procedures were understood and structured correctly. Once the flow charts were approved by the industry contact, the business processes of NADRA customer services was thoroughly understood, based on this information the knowledge trees for the UCA were constructed.

4.6.3 Knowledge Tree Construction

Raux and Eskenazi (2004), state that for meaningful, natural interaction with the user, dialogue/CA systems must follow a model of human task- oriented dialogue. Subsequent to the formal verification of the process flowcharts by the industry contact in NADRA (see Appendix I – verification email). The flowcharts were used to create knowledge trees using the "ConvAgent tree tool" (see appendix F ConvAgent Tree Tool). The knowledge trees were used as a reference to script possible user utterances at each stage of the conversation. Each node of the knowledge tree represents a point within the conversation related to the context of the problem domain; the nodes also highlighted the variables that needed to be captured at each point during the discussion in order for the conversation to be able to reach its goal, which is represented by the leaf node. The decision trees highlighted the conversation paths that had to be followed in order for the conversation to reach its goal.

The knowledge base for the prototype UCA consists of 3 main layers/ that are: the domain specific layer which is the ID card application layer, Frequently Asked Questions Database (FAQDB), and a general conversation layer illustrated in Figure 32. Each layer represents a context, and each context has all the related sub contexts mapped to it. Each context represents a state of the discussion the UCA can be in; from this the UCA is able to determine what the user wants from the discussion and also allow the CA to be aware of the context/topic of the discussion. The ID card layer is a domain specific layer it holds all the sub-contexts related to ID card application, this is the main layer that holds all the knowledge relate to the domain. Each sub context in this layer relates to a different area of discourse within the domain and captures different attributes depending on the context. Within each layer all the sub contexts related to that state are mapped together and each sub context has a pre-defined path that is linked together using the knowledge tree logic. Each path leads to a leaf node in the knowledge tree, the conversation follows the knowledge tree logic until it reaches a leaf node which is the goal of the conversation. The FAQ layer is there to handle frequently asked questions related to the domain, for example how much does it cost to apply for an ID card, how much it cost, which application from is required etc. The list of FAQ was also compiled from the interview with the industry contact. The general layer deals with general conversation not related to the domain, such as greeting and everyday “small talk” (i.e. the weather, sport and politics etc.). The general layer is included in to the knowledge base to make the UCA seem more intelligent as CA’s should have responses that are not related to the main domain to seem more intelligent. However as it is not possible to cover all aspects of general talk a few select sub-contexts have been implemented.

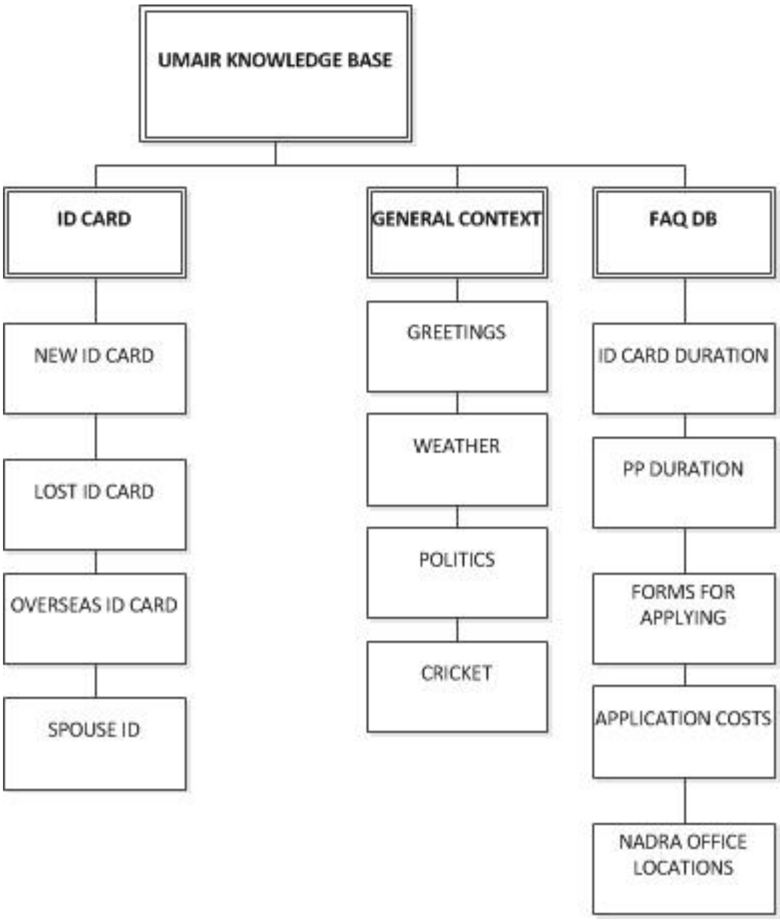


Figure 32 - Domain structure

4.6.4 Conversation Design

When designing the conversation the scope of the domain was the first aspect to be considered, in terms of length (number of utterances) and the desired goal/outcome to be achieved through the dialogue. The processes of ID card application and stages involved dictated the length of the conversation as different ID card related queries involved different steps. Based on this the conversation flow was designed and structured as a set of rules called a conversation path. Each path was stored in sequence in the CA database (see path manager section 4.5.8). Each path is responsible for a different sub topic within the context of the problem as illustrated in Figure 33. The last rule of the path is the goal of that path, when the user arrives at this rule the objective/goal of the conversation is achieved.

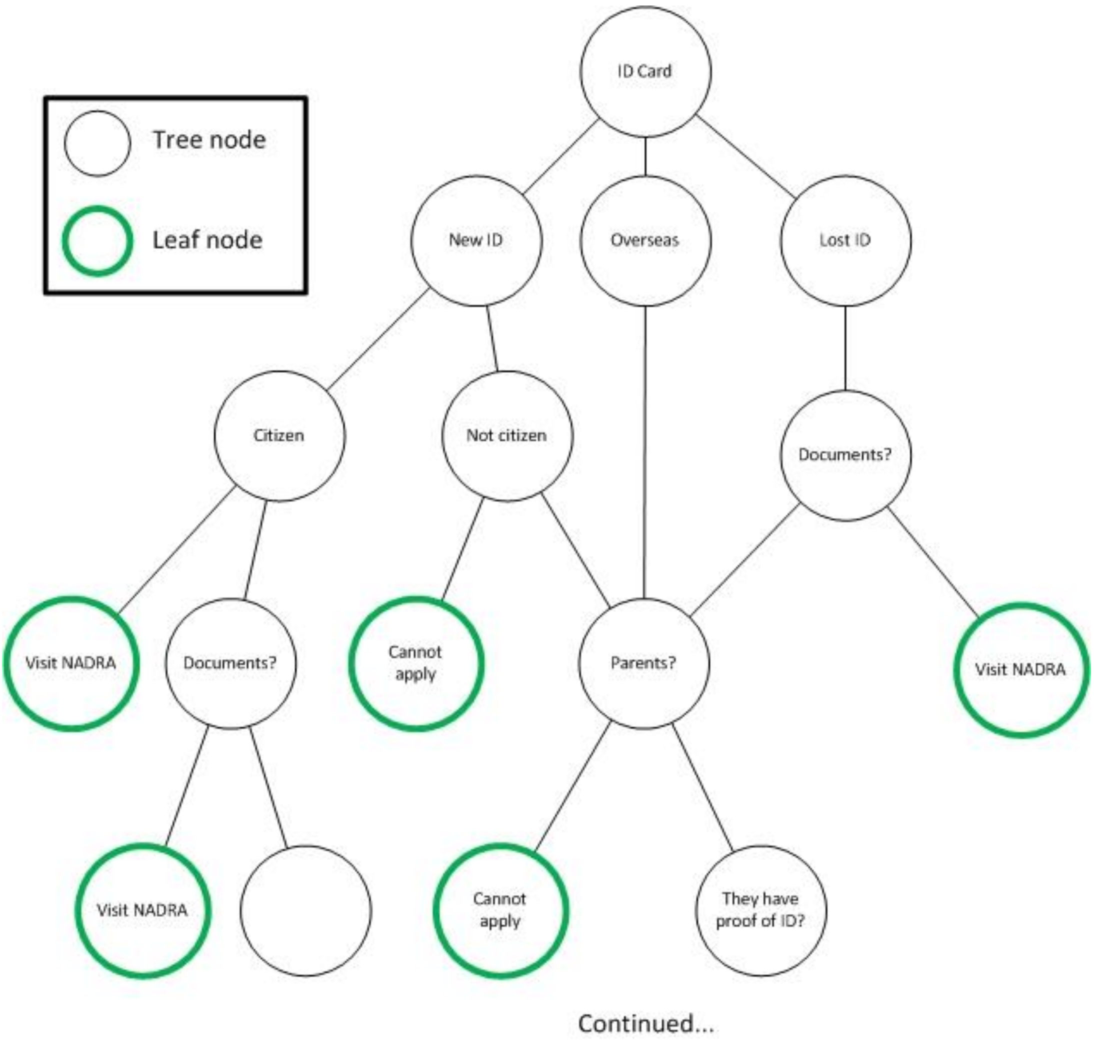


Figure 33 - Example of sub contexts mapped within context layers

The design of the conversation was a time consuming and iterative process, however planning and detailing the dialogue at this point, resulted in the development of the conversational agent to be more efficient.

4.6.5 Strategies for conversation

The strategies for the conversation were decided based on the interviews with the industry contacts. UMAIR is designed to mimic a customer service representative, therefore the use of unacceptable/inappropriate language during customer interaction is inevitable. This notion is supported by Grandey et al. (2004) who suggest that verbal forms of aggression (e.g., yelling, insults, and cursing) are the most frequently experienced forms of aggressive behaviours in a customer facing role. To handle this type of behaviour there is a three strikes rule implemented within UMAIR. The three strikes rule deals with inappropriate language by allowing the user three chances if bad/unacceptable language is used. If UMAIR recognises that unacceptable language has been used by the user, the user is given a warning that if that type of language persists the conversation will be terminated. If at any time during the conversation the user uses up the three chances the conversation is terminated.

If the UMAIR is not able to understand a user utterance UMAIR responds to the user stating that the last utterance was not understood. In an instance where the user converses out of context during a discussion (i.e. something not related to the domain or context of the discussion) UMAIR checks the out of context utterance for a match in all of the knowledge base layers (general discussion, frequently asked questions etc.) and delivers and appropriate response then directs the user back to the point where the conversation digressed from.

4.7 Phase 4: Testing and Evaluation of the UCA

The final phase involved thorough testing of all the developed components. Through carefully designed experiments explained in detail in the subsequent chapter.

4.8 Chapter Summary

This chapter has detailed the methodology and implemented components which comprise the UMAIR prototype engine. Due to the nature of the Urdu language and the current state of Urdu language research it was not feasible to create an Urdu conversation using existing CA development methodologies and components. In light of this several key components that deal with the language unique challenges of Urdu (e.g. WOW algorithm, scripting language, utterance filter etc.) have been researched, developed and implemented in UCA/UMAIR’s architecture. These components form the architecture of the first UMAIR prototype which will be tested for their effectiveness and robustness in order to gather evidence to answer the main research question of 'can the Urdu language be implemented in a CA to produce an effective, functional CA?' The testing/evaluation methodology, experiments and results are detailed in the ensuing chapters.

Chapter 5 - UMAIR Prototype 1 – Evaluation Methodology and Results

5.1 Introduction

This chapter reports the evaluation methodology and results of the evaluation of the first UMAIR prototype conducted to gauge the effectiveness of the framework and architecture. In chapter four a novel Urdu CA framework was proposed, which is designed specifically to address several key challenges posed by implementing Urdu in to a CA. The framework was utilised to implement UMAIR, UMAIR is a goal orientated CA which is design to emulate a NADRA customer service advisor. The architecture of UMAIR incorporates novel components such as the WOW similarity algorithm and scripting language. In order to validate the implemented Urdu CA framework and the UMAIR architecture proposed in chapter 4 section 4.3 an empirical study was undertaken in a real world domain to evaluate whether the Urdu language with all its complexities could successfully be implemented into a CA, and if the resulting CA can deliver an effective conversation and help the user to reach their desired goal through discourse. The preliminary evaluation aims to evaluate the effectiveness, functionality and robustness of UMAIR architecture and components. In order to shed light on the following points:

- The accuracy and robustness of the PM and WOW similarity algorithm.
- Can UMAIR closely imitate a human NADRA representative in reasoning, logic and information given and conduct a conversation by leading and directing the user towards the goal of the conversation?
- Can UMAIR converse in Urdu, recognise the Users requirements and guide them towards the goal of the conversation?
- Do the developed framework and architecture components address the challenges of the Urdu language?

Since there is no standard framework/approach available for evaluation of CA's, a novel evaluation framework has been developed which focuses on evaluating UMAIR from subject and objective perspectives. Initial experiments are conducted to evaluate the effectiveness and functionality of the prototype agent from the subjective perspective, as perceived by the sample user group. As well as this the UCA is tested from an objective perspective through the capture and analysis of key conversation related metrics which are utilised to gauge system robustness, conversation success/task completion and effectiveness of the new WOW algorithm.

A total of 24 participants are recruited to evaluate UMAIR, through a scenario based evaluation strategy. The scenario's all related to the domain of the NADRA ID card application. The participants selected had to be fluent readers and writers of Urdu and English. It was difficult task to convey the scenario to the participants since any Urdu description would cause bias in the language they would use to interact with the system. Therefore to mitigate this, each scenario was explained in English and the participants were asked to interact with UMAIR in Urdu. Since English and Urdu are in entirely different language families, this kind of design should minimise such bias.

In addition to this a Wizard of Oz (WOZ) (Maulsby et al., 1993) style experiment is conducted in order to gauge if the participants perceived any significant differences between UMAIR and a human presented through the agents interface as the Wizard in the WOZ experiment.

The following sections detail the evaluation experiments conducted with participants and the metrics that are to be measured through the experiments. The results of the experiments are statistically analysed and presented.

5.2 Experiment Methodology

The initial prototype UCA is evaluated through experiments designed to measure different aspects of the CA. The experiments conducted are used to test the effectiveness of the conversation, end user satisfaction, usability and system robustness. For the initial prototyping stage, gauging these aspects will highlight areas for system improvement and which can be addressed through further research.

The effectiveness of the experimental design outlined in this chapter is tested through a pilot study conducted with a small number of participants to evaluate the design of the full-scale experiment. The results of the pilot study were then utilised to adjust the full scale experiment to ensure its accuracy. A pilot study is a valuable insight and can highlight discrepancies in the experimental design (e.g. questionnaire questions, or log file contents). The experimental design can then be adapted to improve the chances of a clear outcome.

5.2.1 Hypotheses

The hypotheses to be tested, relate to the effectiveness of the UCA prototype system. They are as follows:

- H1 – UMAIR and WOZ user perceptions are not equal
    - The grammatical and lexical challenges involved mean that it is not possible to produce an effective functional Urdu CA (i.e. the users perceive a statically significant difference between UMAIR and WOZ).
- H0 – UMAIR and WOZ user perceptions are the equal
    - The novel engine, scripting language and methodologies deployed result in an effective functional Urdu Conversational Agent. (i.e. the users do not perceive a statistically significant difference between UMAIR and WOZ).

To test these hypotheses the GQM model (section 2.9.1) is utilised to formulate which metrics are required to be measured in order to gather data to test them.

The UMAIR GQM model illustrated in Figure 34 outlines a number of metrics that are required to successfully test the hypotheses. The goal of the GQM model is based on the main research hypothesis of researching and developing an effective Urdu conversational agent. The relating questions are the questions that relate to the effectiveness and quality of a software application, which take into consideration the objective and subjective perspectives of the perception of quality. The metrics identified are a mixture of subjective and objective metrics which need to be gauged in order to answer the questions.

The robustness of UMAIR is mainly evaluated through objective metrics related to the conversation. For example the number of correct and incorrect answers will gauge whether or not the novel engine is able to handle the challenges of Urdu such as free word order, diacritical marks and still be able to deliver a coherent conversation to the end user. Accordingly the number of unrecognised utterances will be measured, this will provide insight in to how effective UMAIR's engine is at mitigating the challenges.

The subjective metrics are related to gauging the perceptions of the user with regards to their experience of interacting with UMAIR. The subjective and objectives metrics to be evaluated and their mode of evaluation are summarised and outlined in Table 4 and Table 5.

5.3 Formulation of Evaluation Metrics

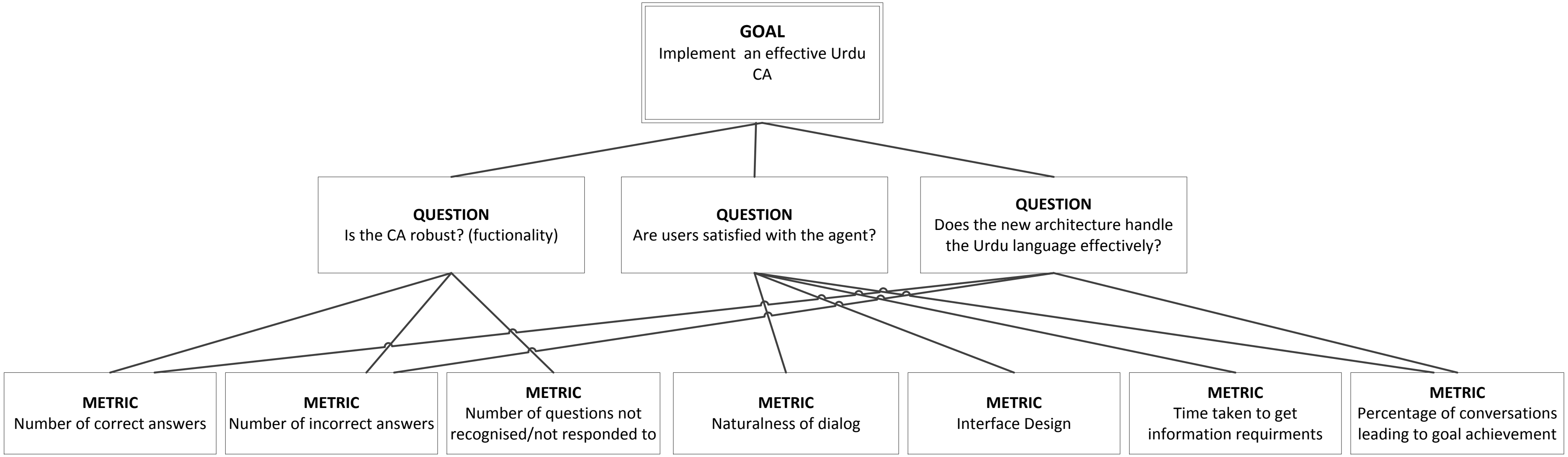


Figure 34 - GQM Model for UCA Evaluation

The metrics are selected based on the GQM methodology detailed in chapter 2 section 2.9.1.

SUBJECTIVE METRICS		
Metric to be Evaluated	Mode of Evaluation	Characteristic Measured
Agent naturalness	Questionnaire	Usability/user satisfaction
User Interface (UI) design	Questionnaire	Effectiveness of the UI/user satisfaction
Time take to get information required	Questionnaire/Log File	Usability
Overall user satisfaction	Questionnaire	Overall effectiveness of the UCA from end users perspective

Table 4 – Subjective evaluation metrics

OBJECTIVE METRICS		
Metric to be Evaluated	Mode of Evaluation	Characteristic Measured
Number of correct responses	Log file	Agent accuracy/robustness
Number of Incorrect responses	Log file	Agent accuracy/robustness
Number of unrecognised utterances	Log file	Agent robustness/robustness
Agents ability to understand user utterances	Log file	Agent robustness/robustness
WOW algorithm	Log File	Algorithms ability to handle word order variation Effectiveness of the similarity calculation
Number unrecognised utterances	Log File	Scripting/robustness
Goal of conversation achieved	Log File	Agent effectiveness/robustness

Table 5 - Objective evaluation metrics

5.4 Data Collection

5.4.1 Subjective Data Collection

The data to ascertain the subjective measures will be gathered through an end user questionnaire. The questionnaire designed is based on the research questions that need to be addressed. With this in mind, the questionnaire is comprised of questions based on an ordinal scale such as the Likert scale (Brooke, 1996). According to Galán et al. (2013) a Likert scale is a psychometric response scale primarily used in questionnaires to obtain a participants preferences or degree of agreement with a statement or set of statements. Respondents are asked to indicate their level of agreement with a given statement by way of an ordinal scale. The Likert scale is a widely accepted tool for researchers to utilise when gathering information related to attitudes, emotions and opinions. Moreover, the Likert scale allows the quantification of subjective metrics that are not directly measurable (Gliem and Gliem, 2003). This method has been utilised by Martinez et al. (2008) and Lutfi et al. (2013) for the evaluation of dialogue systems through end user interaction. The data gathered through the questionnaire will help evaluate the subjective measures of the UCA.

5.4.2 Objective Data Collection

The data to measure the objective measures will be derived from the log file generated from the UCA system, which records discourse related metrics about the users discussions. The log file will be analysed subsequent to the user's interaction. The log file will provide backend insight into the workings and success of the system and its associated algorithms. Each participant will automatically be assigned a unique session ID by UMAIR once they start to use the system. The session ID will then be utilised to identify and analyse each of the participant's conversations with UMAIR.

5.5 Participants Sample & Demographic

The total size of the sample consisted of 24 participants. Participants for the evaluation were difficult to locate as the researcher sought to recruit participants who were fluent in Urdu and English readers and writers (Li and Jagadish, 2014). The reason for targeting this particular demographic is because, subsequent to interacting with UMAIR in Urdu the participants will be asked to fill in a feedback questionnaire in English about their experience and perceptions. The participants recruited for the evaluation are residents of the Greater Manchester area. They are all native Urdu speakers and fluent in English and could read and write in both languages. The participants spanned varying age groups (18 - 50) and education levels and both genders are represented in the sample. None of the participants involved in the evaluation/testing will have any previous experience using UMAIR. The participants were not paid for their participation in the evaluation study they all volunteered to participate for altruistic reasons.

5.6 Evaluation Scenarios

Scenarios have numerous possible applications in system development (Alexander and Maiden, 2005, Carroll, 1995). Carroll (1995) highlights many different applications of scenarios in the system development lifecycle. One such application is the use of scenarios in the evaluation and testing phase of software development. Carroll (1995), states since "a system must be evaluated against the specific user tasks it is intended to support", scenarios are ideal for usability evaluation.

Scenario-based evaluation methods evaluate software's ability with respect to a set of scenarios of interest which are derived from the goals of the developed software. Scenario is brief descriptions of a single interaction of a stakeholder/participant with a system (Roy and Graham, 2008). All the pre-defined evaluation scenarios given to the participants were devised through the data acquired through the interviews with the industry contact in the knowledge engineering phase. The scenarios are all based on real world queries collected through the knowledge engineering stage (section 4.6.2), and are all scenarios that the NADRA department receive and deal with on a daily basis devised through the interviews with the industry contact. The scenarios were a mixture of complex and simple tasks related to the domain (see Appendix E for a detailed list of scenarios). The scenarios were given to the participants as a guide to their interaction with UMAIR.

5.7 Participants Interaction

5.7.1 Experiment 1 – An experiment in a Wizard of OZ setting

The aim of this experiment is to highlight the strengths and weaknesses of the developed UMAIR architecture. The experiment will test H1 through a Wizard of Oz experiment. According to Wilson and Rosenberg (1988), Wizard of Oz is a rapid-prototyping method for systems costly to build or requiring new technology. A human "Wizard" simulates the system's intelligence and interacts with the user through a real or mock computer interface. A wizard is then able to select an appropriate response from a set of previously defined utterances or use a free-text field to compose a response on the fly (Schlögl et al., 2014).

In a Wizard of Oz experiment evidence for H1 will take the form of the agent not being distinguishable from the human foil in terms of objective task completion criteria. The wizard will be a human participant who will be sufficiently versed in the domain knowledge and will be given a printed version of the decision tree deployed in UMAIR's engine. The Wizard will respond to the user based on the logic set out in the decision trees. Thus domain expertise is controlled as a confounding factor and the experiment specifically evaluated UMAIR's conversational abilities.

5.7.1.1 Experimental Methodology

The participants were verbally briefed prior to their interaction with the system that the system is a prototype and that it can only answer questions about one area of the domain (i.e. ID card application). The participants were instructed to interact with the system as they would if it were online. They were also will also be instructed that the scenarios were only guidelines to specify the possible tasks that the agent could address and that they were free to go ahead and interact with the system as they felt appropriate (e.g. language used). Meetings were arranged with all the participants and most of the evaluations took place over two days

The participants were given their particular problem/scenario related to the domain prior to them using the system (see Appendix E for list of scenarios), and are instructed to ask the UCA how to solve their particular problem. Similar methods for dialogue system evaluations have been used by Martinez et al. (2008), Lutfi et al. (2013) and Janarthanam et al. (2013) . The participants are not be informed whether they are speaking to a wizard or the prototype system. The human wizard will provide an answer based on the decision trees which are deployed in the prototype system. The knowledge trees will be printed and given to the human participant who will play the part of the "wizard" in all of the experiments. The part of the wizard was played by a friend of the researcher who was familiar with the domain.

Subsequent to participants interacting and engaging with each system to complete their scenario (UMAIR and the WOZ), they fill out a user experience questionnaire (see Appendix A – Evaluation questionnaire) which will ask them to rate their experience and opinions about their subjective perceptions of interacting with the systems. Participants were asked to identify whether they thought that the tasks were successfully completed as well as other questions related to measuring the subjective metrics (see Table 4, section 5.3 for list of subjective metrics). The results of both of the questionnaires will be analysed and compared to establish whether or not the participants perceived a significant different between UMAIR and the WOZ.

5.7.2 Experiment 2 – Log file analysis of experiment 1 data

The aim of this experiment is to test the robustness of the UMAIR architecture and its corresponding components. The data for this experiment will be gathered from the UMAIR's log file (Appendix C – Excerpt from UMAIR log file) subsequent to the end user evaluation outlined in the previous section. This experiment will provide supplementary objective data in order to test H1. The log files of both of UMAIR and WOZ will be analysed and compared, in order to gauge the success UMAIR in task completion and effectiveness compared to the WOZ.

5.7.2.1 Experimental Methodology

The data gathered in the log file memory will allow key insight into the performance of UMAIR and the algorithms deployed in the architecture. This data will help to gauge the success of the objective metrics for example the robustness of the system and the effectiveness WOW similarity algorithm (see Table 5, section 5.3 for list of objective metrics) and other architecture components. The log file from the WOZ will also be utilised to provide further insight in to the participant's conversations which can be utilised to expand the knowledge base and address gaps in the scripting.

5.8 Experimental Data Analysis

The data gathered from the participant interactions in experiment 1 (i.e. log files) will be collated, tallied, and subsequently analysed to explore the findings from experiments 1 and 2. Software packages such as SPSS and Microsoft Excel will be utilised where necessary to display and analyse the data. The data that will be analysed will be the questionnaire results (subjective), and the log file data (objective) collated during the participants interaction with UMAIR and the WOZ for all 24 participants. The data will be analysed in accordance to the type of data collated (i.e. parametric non-parametric). The data will highlight weaknesses and areas for improvement within the system.

5.9 Chapter Summary

This chapter has outline and detailed the methodology utilised to evaluate the prototype UMAIR CA. The methodology aims to evaluate the system from objective and subjective perspectives. The evaluation will highlight the strengths and weaknesses of the system which will be addressed through further research. The following chapter details and analyses the results of the end user evaluation.

Chapter 6 - Evaluation Results and Discussion

6.1 Data Reliability

According to Foster (2001), reliability refers to the consistency of the results on different items in the test. To understand the relationship between different items of data, it is necessary to quantify the reliability of the data. Prior to undertaking a detailed analysis of the data, each variable contained within the research scale was tested for reliability. In this case a variable is any Likert scale question of the questionnaire. In this way the measurement device is tested (Hammond, 1995). If the reliability was found to be low the credibility of the outcome would need to be questioned. Accordingly, the internal consistency of the measurement scale was evaluated with the use of Cronbach's coefficient (Coakes and Steed, 2001).

One of the most commonly used indicators of internal consistency is Cronbach's alpha coefficient. This statistic provides an indication of the average correlation among all of the items that make up the questionnaire scale. Values range from 0 to 1, with higher values indicating greater reliability. Ideally, the Cronbach alpha coefficient of a scale should be equal to or above .7 (Pallant, 2004). Table 6 displays the results of the Cronbach alpha test conducted to test the questionnaire scale from experiment 1.

	Cronbach's Alpha
Participants perception of UMAIR's Helpfulness	.850
Participants perception of the WOZ Helpfulness	.826
Participants perception of the information and instructions given by UMAIR	.828
Participants perception of the information and instructions given by WOZ	.842
Participants perception of UMAIR level of understanding	.818
Participants perception of WOZ level of understanding	.822
Participants perception of the naturalness of the conversation with UMAIR	.844
Participants perception of the naturalness of the conversation with WOZ	.856
Participants level of satisfaction with UMAIR	.836
Participants level of satisfaction with WOZ	.825
Participants perception of the time taken to complete the conversation with UMAIR	.853
Participants perception of the time taken to complete the conversation with WOZ	.851

Table 6 - Cronbach alpha test of reliability

The results in Table 6 reveal that each scale in the evaluation questionnaire has good internal consistency, with a Cronbach alpha coefficient greater than .8 reported for all the questions. This indicated the data gathered is giving a reliable and consistent picture of each attribute.

6.2 Rationale for the selection of statistical test

Choosing the right statistical technique for data analysis is the most difficult part for any research (Pallant, 2004). One such reason highlighted by Kinner and Gray (2000) is that there is no universal methodology to help researchers to choose the right statistical test. It is the variations in the types of research that makes the selection of right statistical test a challenging task. Selecting the right statistical tests depends on the sort of research questions that need to be answered, the scale utilised in questionnaire, the variables to be analysed, the assumptions met by the data for specific statistical techniques, and the nature of data itself (Pallant, 2004).

In statistics there is often reference to two different types of statistical techniques: parametric and non-parametric. The word parametric comes from parameter, or characteristic of a population. The parametric tests (e.g. t-tests, analysis of variance) make assumptions about the population that the sample has been drawn from. This often includes assumptions about the shape of the population distribution (e.g. normally distributed). Non-parametric techniques, on the other hand, do not have such stringent requirements and do not make assumptions about the underlying population distribution (which is why they are sometimes referred to as distribution-free tests). Non-parametric statistics are inferential statistical analyses designed to be used when the data is not normally distributed and not based on a set of assumptions about the population (Nolan and Heinzen, 2011) this most often means they are used with categorical and ordinal data. In contrast, parametric statistics are inferential statistical analyses based on a set of assumptions about the population and require numerical score (Gravetter and Wallnau, 2002).

Normal distribution can be checked observing the histograms, by checking the ratio of skewness and standard error, or by ratio of kurtosis and standard error, and also by performing the test of normality. ‘Tests of normality’ is the other option to ascertain normality and can be done by using Kolmogorov-Smirnov test for a sample size greater than 50 or Shapiro-Wilk test if sample size is smaller than 50 (Gravetter and Wallnau, 2002, Nolan and Heinzen, 2011, Pallant and Manual, 2010). The convention is that significant values greater than 0.05 indicates that sample scores are similar to a normal distribution. The histograms for the questionnaire results are shown in Appendix G along with the results of the normality test shown in Table 7.

Q		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Statistic	df	Sig. (p)	Statistic	df	Sig. (p)
1	Participants perception of UMAIR's Helpfulness	.464	24	.000	.542	24	.000
	Participants perception of the WOZ Helpfulness	.409	24	.000	.677	24	.000
2	Participants perception of the information and instructions given by UMAIR	.414	24	.000	.689	24	.000
	Participants perception of the information and instructions given by WOZ	.410	24	.000	.710	24	.000
3	Participants perception of UMAIR level of understanding	.414	24	.000	.689	24	.000
	Participants perception of WOZ level of understanding	.411	24	.000	.636	24	.000
4	Participants perception of the naturalness of the conversation with UMAIR	.427	24	.000	.647	24	.000
	Participants perception of the naturalness of the conversation with WOZ	.401	24	.000	.616	24	.000
5	Participants level of satisfaction with UMAIR	.291	24	.000	.788	24	.000
	Participants level of satisfaction with WOZ	.253	24	.000	.856	24	.003
6	Participants perception of the time taken to complete the conversation with UMAIR	.375	24	.000	.688	24	.000
	Participants perception of the time taken to complete the conversation with WOZ	.215	24	.006	.887	24	.011

a. Lilliefors Significance Correction

Table 7 - Test of Normailty

The histograms and the Shapiro-Wilk test shows that the data is not normally distributed, hence non-parametric tests will be utilised to analyse the data. The Wilcoxon test will be utilised, it is the non-parametric alternative to the repeated measures t-test, but instead of comparing means the Wilcoxon converts scores to ranks and compares them between the two systems. If the significance level (p-value) is equal to or less than .05 (e.g. .04, .01, .001) then you can conclude that the difference between the two scores is statistically significant (Pallant, 2004).

The data classification for the questionnaire data collated through experiment 1 in terms of normality and reliability has been establish through the above tests. The next step was to further investigate by employing inferential statistical analysis techniques to test if the participants/users perceived a difference between their experiences with UMAIR and the WOZ. The WOZ testing would provide a benchmark of a human conducting the role of a NARDA customer service agent. This data can then be compared to the end user evaluation data from the UMAIR CA to ascertain the effectiveness and robustness of the implemented system and architecture compared to a human.

6.3 Experiment 1 – Results and Discussion

A Wilcoxon Signed Ranks Test was conducted on each pair of corresponding questionnaire questions administered after the participants interacted with UMAIR and the WOZ during experiment 1. The results of the analysis are shown in Table 8. Table 9 outlines the means of each pair of corresponding questionnaire questions.

Test Statistics[c]			
Wizard of OZ	UMAIR	Z	Asymp. Sig. (2-tailed)
Participants perception of the WOZ Helpfulness -	Participants perception of UMAIR's Helpfulness	-.447[a]	.655
Participants perception of the information and instructions given by WOZ -	Participants perception of the information and instructions given by UMAIR	-1.342[b]	.180
Participants perception of WOZ level of understanding -	Participants perception of UMAIR level of understanding	-1.414[a]	.157

Participants perception of the naturalness of the conversation with WOZ -	Participants perception of the naturalness of the conversation with UMAIR	-2.236[a]	.025
Participants level of satisfaction with WOZ -	Participants level of satisfaction with UMAIR	-1.732[b]	.083
Participants perception of the time taken to complete the conversation with WOZ -	Participants perception of the time taken to complete the conversation with UMAIR	-3.882[b]	.000

a. Based on negative ranks.

b. Based on positive ranks.

c. Wilcoxon Signed Ranks Test

Table 8 - Wilcoxon Signed Ranks Test Results

Paired Means Statistics

	Mean	N
Participants perception of UMAIR's Helpfulness	3.75	24
Participants perception of the WOZ Helpfulness	3.79	24
Participants perception of the information and instructions given by UMAIR	3.88	24
Participants perception of the information and instructions given by WOZ	3.75	24
Participants perception of UMAIR level of understanding	3.88	24
Participants perception of WOZ level of understanding	3.96	24
Participants perception of the naturalness of the conversation with UMAIR	3.17	24
Participants perception of the naturalness of the conversation with WOZ	3.38	24
Participants level of satisfaction with UMAIR	3.79	24
Participants level of satisfaction with WOZ	3.67	24
Participants perception of the time taken to complete the conversation with UMAIR	4.00	24
Participants perception of the time taken to complete the conversation with WOZ	2.58	24

Table 9 - Paired means

The first test was performed to find out whether there were differences in the **perception of helpfulness** between the UMAIR and WOZ. The results are considered to be significant, with a value of $p < 0.05$. It can therefore be concluded from Table 8 that there is no statistically significant difference between perception of helpfulness between the UMAIR and WOZ, $p = .655$, indicating that **H0 can be accepted**.

Another test was conducted to find out whether there was a significant difference in the scores for users **perception of quality of information and instructions** between the UMAIR and WOZ. It can therefore be concluded from the results of the Wilcoxon test in Table 8 that there is no statistically significant difference between perception of quality of information between the UMAIR and WOZ, p = .180, meaning that **H0 can be accepted**.

Further tests have been carried out to find out whether there was a significant difference in the scores for users **perception of the level of understanding** between the two systems. It can therefore be concluded from the results of the Wilcoxon test in Table 8 that there is no statistically significant difference between the users perception of the level of understanding between the UMAIR and WOZ, p = .157, meaning that **H0 can be accepted**.

A further test was conducted out to find out whether there was a significant difference in the scores for users **perception of the naturalness of the conversation** between the two systems. It can therefore be concluded from the results of the Wilcoxon test in Table 8 that there is a statistically significant difference between the users perception of the conversation naturalness between the UMAIR and WOZ, p = .025, meaning that **H1 can be accepted**. The mean scores in each case are UMAIR (M= 3.17) and WOZ (M= 3.38). The total means of the two scores highlight that the participants perceived their conversation with the WOZ as more natural.

An additional test was conducted out to find out whether there was a significant difference in the scores for users **level satisfaction with the conversation** between the two systems. It can therefore be concluded from the results of the Wilcoxon test in Table 8 that there was not a statistically significant difference between the users perception of the level satisfaction with the conversation between the UMAIR and WOZ, p = .083, meaning that **H0 can be accepted**.

The final test was conducted out to find out whether there was a significant difference in the scores for **the users satisfaction level with time taken to complete the conversation** between the two systems. It can therefore be concluded from the results of the Wilcoxon test in Table 8 that there was a statistically significant difference between the time taken to complete the conversation naturalness between the UMAIR and WOZ, p = < .000, meaning that **H1 can be accepted**. The mean scores in each case are UMAIR (M= 4.00) and WOZ (M= 2.58). From the mean scores it can be observed that UMAIR received a significantly better response as to the time take to complete the conversation.

The results demonstrate that out of the six metrics tested to gauge the effectiveness, in four areas (helpfulness, quality, understanding and level of satisfaction) the users/participants could not perceive any statically significant difference between their conversation with UMAIR and the WOZ. In two areas the users/participants did perceive a statically significantly difference these two areas were the naturalness of the conversation and level of satisfaction with the time taken to complete the conversation.

In addition the frequency analysis of the questionnaire results illustrated in Table 10 highlights that the participants disliked the interface of UMAIR. Upon reading the comments section of the questionnaire the general consensus among the participants with regards to the interface was that is looked too plain and boring and not interactive.

QUESTIONNAIRE RESULTS UMAIR					
	SCALE				
QUESTION	1	2	3	4	5
1. Design	4.2%	41.7%	54.2%	-	-
2. Helpfulness	-	-	25.0%	75.0%	-
3. Information & Instruction	-	4.2%	12.5%	75.0%	8.3%
4. System Understanding	4.2%	-	12.5%	75.0%	8.3%
5. Naturalness	4.2%	-	75.0%	20.8%	-
6. Satisfaction	-	-	33.3%	54.2%	12.5%
7. Did the system help with your query?	-	-	12.5%	75.0%	12.5%
	YES			NO	
8. Use the system again?	96%			4%	
9. Use instead of human?	42%			58%	

Table 10 - Frequency analysis UMAIR questionnaire

An additional finding from the results indicated that the vast majority (96%) of the user/participants said that they would use the system again, however less than half of the respondents felt that they would use the system in preference to a face to face conversation with a human.

6.4 Experiment 2 – Results and Discussion

The purpose of experiment 2 was to further gauge the effectiveness and robustness of UMAIR compared to the WOZ in order to test hypothesis 1 from an objective perspective. The data from both UMAIR's log file and the log file of the WOZ experiment was collated coded and analysed. An example translated transcript is illustrated in Figure 35. The participant in the example was given scenario 1, which was to ask UMAIR how to obtain a new ID card, as a first time applicant.

		English translation	Urdu Utterance	Analysis
1	P	I need a new ID card	شناکتی کارڈ مجھے بنانا ہیں	Initialisation rule fired for new ID card context
2	U	Are you a citizen of Pakistan?	کیا آپ کا تعلق پاکستان سے ہے ؟	
3	P	Yes, I am a citizen of Pakistan	جی ہاں میں پاکستانی ھوں	Sentiment classified as positive (decision tree)
4	U	Have you ever had an ID card before?	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	
5	P	No, this will be my first	نہیں یہ میرا سب سے پہلے ہے	Sentiment classified as negative (decision tree), utterance is word order variation of scripted pattern
6	U	OK, then do you have any of the following document to prove your citizenship…..	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	Capture variables from next utterance. Which document?
7	P	Yes I have a birth certificate	ہاں جی میرا پاس پیدائش کا سرٹیفکیٹ ہیں	Sentiment classified as positive. (decision tree) Has document: Birth Certificate. (Variable captured)
8	U	Ok, have you filled out the POC form?	کیا اپ نے فارم مکمل کر دیا ہ؟.	
9	P	yes that form is completed	جی ہاں فارم مکمل ہیں	Sentiment classified as positive (decision tree)
10	U	Have you deposited the application fee to the bank?	کیا اپ نے دروست فیس درخواست کے ساتھ لگا دے ہے یا بینک میں جمع کروا دے ہے؟	
11	P	Yes I have	جی ہاں	Sentiment classified as positive (decision tree). Utterance is word order variation of scripted pattern.
12	U	Ok in that case you have fulfilled all the requirements, please send your application to your local NADRA office.	اپ اپنی کارڈ درخواست قریبی نادرا دفترمیں کرا دیں	Conversation goal achieved.

Figure 35 - Excerpt from UMAIR Log file transcript

Figure 35 illustrates how UMAIR's engine processes user utterances in order to find matches to the scripted patterns in the knowledge base. The conversation is initiated by the user through initialisation utterance, which is matched to one of the initialisation rules stored in the knowledge base. The firing of the initialisation rule sets the context of the discussion and UMAIR leads the conversation with the user by asking leading questions towards the goal of that particular context. Throughout the discussion the user utterances are processed and check for sentiment, the sentiment analysis is used by the engine to make decisions regarding the knowledge trees, to determine the path of the conversation (e.g. utterance 3, 5, and 7). The example conversation also demonstrates the WOW algorithm and how it is able to recognise user utterances that are word order variations of scripted patterns (e.g. utterance 5). UMAIR was able to recognise that the user utterance: یہ میرا سب سے پہلے ہے* was a variation of the scripted pattern: میرا سب سے پہلے ہے یہ * and correctly respond to the user.

The results of the log file analysis from UMAIR and the WOZ are illustrated in Table 11.

CONVERSATION ANALYSIS		
METRIC	UMAIR	WOZ
Total number of utterances in all conversations	212	219
Average number of words per user utterance	5.0	5.8
Average number of utterances per conversation	8.8	9.1
Average conversation duration (mins)	3.2	13.0
Number of unrecognised utterances	12%	-
Percentage of conversations leading to goal	83.3%	100%
Percentage of utterances containing word order variations of scripted patterns	33.6%	-
Percentage of conversations which reached goal without deviating the context	87%	-

Table 11 - Umair log file analysis

6.5 Discussion

The results in Table 11 reveal that in general UMAIR performed well in comparison to the WOZ. The results show that the users tended to use the same amount of words and marginally less utterances/number of turns during their conversation with UMAIR when compared to the WOZ. The conversations with UMAIR were significantly shorter in time when compared to the conversations with the WOZ. This is also reflected in the opinion of the participants in the end user questionnaire who perceived their conversation with UMAIR led to a quicker/more efficient solution to their problem compared to the WOZ. A reason for this is it took the human Wizard longer to respond, due to the fact that the human had to manually navigate through the decision tree in order to generate the appropriate response. These results are similar to the findings of Skantze and Hjalmarsson (2013) who state "that a common problem with this kind of setting/testing is the time it takes for the Wizard to manage the task (such as transcribe what the user is saying), which may result in long response delays".

Moreover, from the conversations that did reach the goal of the discussion, 87% of them reached the goal without the user going out of context, meaning the user stayed with the context of discussion. However, all of the conversations that strayed away from the context/topic of discussion (13%) did eventually reach the intended goal. From the results, it can be seen that the majority of the participants went straight through their discussion with UMAIR towards the goal. It could be interpreted from these results that the users did not enjoy or like speaking to UMAIR, however on the contrary the questionnaire results highlighted that the users did enjoy their interaction with UMAIR and a vast majority stated they would use the system again (Table 10). It could be said that the users treated UMAIR just like any other customer service representative and just wanted to get the information they required as quick as possible with the least effort.

An additional insight provided from these results is that the WOW algorithm is allowing the reduction of scripted patterns. The results reveal that a third (33.6%) of all the utterances input by the users were actually word order variations of scripted patterns. The log file reveals in total 71 unique utterances relating to 11 different rules contained valid Urdu word order variations which were correctly recognised and dealt with by the WOW algorithm by firing the appropriate rule.

The results also brought to light some of the weaknesses in UMAIR's architecture, mainly the number of unrecognised utterances and the percentage of conversations leading to the goal of the discussion. The WOZ was able to recognise all the utterance the users entered into the system, however in comparison UMAIR failed to recognise some utterances from the users/participants (12%). Upon further analysis of the log file it was found that some of these unrecognised utterances were due to minor spelling mistakes in the user/participants utterances. These spelling errors resulted in the WOW similarity algorithm failing to recognise the word, which meant the match strength of that utterance, was lowered below the acceptable threshold set within UMAIR's engine. Moreover it was evident from the results that word segmentation issue was another aspect that caused the engine to fail to recognise users utterances (chapter 3 section 3.8: details word segmentation issues).

The spelling and word segmentation issues in the user utterances led to misunderstanding and repetition from UMAIR, as UMAIR is programmed to tell the user that 'he didn't understand' and could they please repeat what they were saying. But if the spelling mistake isn't corrected, or the words properly entered (i.e. with spaces) by the user, UMAIR again fails to 'understand' the utterance and the user is prompted again. This led to 3 out of the 4 failed conversations ending due to the users giving up through frustration. It is evident that the spelling mistakes/common spelling variations and inconsistent word segmentation features found in the Urdu language are hampering the accuracy and effectiveness of the UMAIR's engine, in correctly recognising utterances. The other cause for the unrecognised utterances was due some gaps exposed in the knowledge base by the users, but these gaps are easily addressed, simply by adding to the knowledge base. However, the spelling and word segmentation issues are issues that require further research in order to develop new approaches that reduce the impact of these language unique issues on the engine.

6.6 Chapter Summary

The preliminary evaluation revealed some key information with regards to the effectiveness, functionality and robustness of UMAIR. To summarise the main findings of the evaluation are as follows:

- The WOW algorithm managed to reduce the number of scripted patterns by an average of 33%
- UMAIR is able to closely mimic a human and conduct a conversation by leading and directing the user towards the goal of the conversation, with 83.3% of all the conversations leading to the goal.
- UMAIR is able to converse in Urdu, recognise the Users requirements and guide them towards the goal of the conversation.

Based on these findings H0 can be accepted as in 4 out of 6 metrics tested the evaluation participants could not perceive any discernible difference between the WOZ and UMAIR.

The preliminary evaluation and testing has also highlighted areas of weakness within certain components of UMAIR's architecture. Further research is required to address the points highlighted through the preliminary end user evaluation. Additional research is required to make components and algorithms within UMAIR stronger and more robust in order to address the shortcomings unearthed during the evaluation. Further work will entail:

1. Further enhancements to the knowledge base and engine will be made based on the results of the end user evaluation to address weaknesses highlighted.
2. Further research will be carried out to improve the naturalness of conversation delivered by UMAIR in order to improve end user perceptions.
3. The preliminary evaluation has highlighted that spelling has a big impact on the strength and effectiveness of the similarity algorithm when users spell a word incorrectly or in a different way. New methodologies will be researched and developed to overcome the spelling variations which are present in the Urdu language. Since there are no Urdu spell checkers in existence, a novel approach will be taken to overcome this problem.
4. The end user evaluation also highlighted that problem of inconsistent word segmentation is one of the major weaknesses of the engine. The word segmentation issue caused a significant proportion of the unrecognised utterances. Possible approaches to address this issue will be researched and new components will be developed and added to the architecture in order to mitigate this language unique feature of Urdu.
5. Research, develop and enhance the WOW similarity algorithm to improve the matching of Urdu text much more efficiently and reduce the number of unrecognised utterances.
6. Investigate interactive elements to the UCA to make it more engaging. The end user evaluation revealed that the users/participant felt that the prototype was too uninteresting in its visual presentation. Further research will be carried

out into CA's to identify which techniques can be used to achieve a friendlier, interactive experience for the end users.

These weakness and further refinements and enhancements will be addressed by further research and development which is detailed in the next chapter.

Chapter 7 - UMAIR with Improved Architecture

7.1 Introduction

The aim at this stage of the research is to further explore and develop UMAIR's architecture in order to address the issues brought to light through the end user evaluation and so to increase/improve the overall effectiveness, accuracy and robustness of UMAIR's engine and enrich the user experience further. The evaluation experiments revealed some positive results for the architecture and components of UMAIR. However, several key language related weaknesses were highlighted through the end user evaluation. The issues that were revealed were mainly due to the morphological nature and grammatical features of the Urdu language. These language specific issues had detrimental effects on the accuracy and robustness of UMAIRs PM engine.

The issues/weaknesses that were made apparent through the end user evaluation are as follows:

- Urdu Language Features
    - Inconsistent word segmentation
    - Common spelling mistakes/variations
- Architecture Features
    - WOW algorithm similarity calculation
- End user perceptions
    - UI design
    - Conversation naturalness

The most noteworthy among the issues that were highlighted was the word segmentation problem and issues related to spelling errors. These two issues combined were responsible for a significant proportion of the unrecognised utterances during the evaluation. Other issues that were identified through the evaluation of the first prototype were the need to further expand and develop the knowledge base in order to increase the naturalness of the conversation, which was an issue expressed by participants via the end user questionnaire. Another point of concern which was revealed through the questionnaire was the UI. The participants expressed that they thought the UI was uninteresting and plain. The research and development decisions made in order to mitigate these weaknesses are as follows:

- WOW Algorithm
    - The WOW algorithm was improved in order for it to recognise and better deal with the common spelling mistakes made in Urdu. This was achieved by adapting one of the algorithms that is used to assign edge weights to the tokens during the similarity calculation process of the WOW algorithm (see section 4.5 for WOW algorithm and section 7.2 for updated algorithm).
- Inconsistent word segmentation
    - In order to address this issue a new Urdu word segmentation algorithm was developed in order to pre-process the utterances and insure the tokenisation process of the utterance produces valid words (see section 7.4).
- Spelling related issues
    - To mitigate the issue of common spelling mistakes and variations a predictive text input feature was added to UMAIRs architecture (see section 7.3).
- Knowledge base expansion
    - The knowledge base was expanded through further knowledge engineering in order for UMAIR to seem more natural during conversation and add more domain specific as well as general knowledge to the database (see section 7.6).
- UI design
    - The UI of the UMAIR was changed to include an embodied character to enrich the user experience and to assist in clarification/disambiguation in an effort to improve end user perceptions related to UI design as well as conversation naturalness (see section 7.9).

These components are detailed in the subsequent sections of this chapter. The combination of these changes contribute to improving the effectiveness and accuracy of UMAIRs engine in terms of objective task completion as well as addressing the weaknesses found in the subjective metrics.

7.2 Improvements to the WOW similarity algorithm

The findings from the first evaluation revealed positive results for the WOW algorithm, in terms of its ability to recognise and process word order variations of scripted patterns and reduce the scripting effort. However there were some points highlighted through the end user evaluation that could be improved in order to make the algorithm more robust and further improve the similarity calculation. It was found that one of the weaknesses that needed to be addressed was the common spelling mistakes made by the users.

As discussed chapter 3 section 3.9, the Urdu language has several common spelling variations/mistakes made by user due to the phonological similarity of some of its alphabet characters (e.g. س (seen) and ص (saad) both represent a sound similar to the letter S in English) these groups of characters are often inadvertently used interchangeably by users. These errors in Urdu are mainly caused due to homophone Characters. Homophone characters are those characters, which represent the same sound. In Urdu, the number of homophone characters is relatively large compared to English (Naseem and Hussain, 2007). Table 12 below shows the groups of characters in the Urdu language which are phonologically similar.

Character	Phonologically similar character	English equivalent
س	ص	S
ک	ق	K/Q
ک	خ	K
ذ/ز	ض	Z
ت	ط	T/TH
ھ	ح	H
ع	ا	A

Table 12 phonologically similar characters

It was found through the log file analysis subsequent to the first evaluation that a significant proportion of unrecognised utterances stemmed from spelling mistakes made by the user. The spelling errors included the substitution of single letter in a word for a letter which is similar in sound. A typical example is found in the Urdu word for ID card, شناکتی instead of شناختی which is the correct spelling. In the example the word has a common spelling error where the letter that represents the K sound is substituted for the other letter in the Urdu alphabet which is similar phonologically (i.e. ک and خ).

Therefore in order to reduce the impact of these commonly mistaken characters have on the similarity calculation, the edit distance component of the WOW algorithm was adjusted. The updated version of the edit distance algorithm is adjusted specifically to deal with this Urdu language issue, the original implementation of the edit distance algorithm is outlined in Chapter 4 section 4.5. The edit distance/similarity component of the algorithm was adapted to compensate for common spelling variations by allowing the substitution of phonologically similar characters without incurring the cost of a substitution illustrated in Equation 7

$$lev_{a,b}\,(i,j) = \begin{cases} \max(i,j) & if \min \begin{matrix} (i,j) = 0, \\ \boldsymbol{phono\ group\ (i,j) = 0}, \end{matrix} \\ \min \begin{cases} lev_{a,b}\,(i-1,j) + 1 \\ lev_{a,b}\,(i,j-1) + 1 \\ lev_{a,b}\,(i-1,j-1) + [a_i \neq b_j] \end{cases} & otherwise. \end{cases}$$

Equation 7 – Updated edit distance algorithm

The edit distance is utilised as the edge weight of each token in the edge weight matrix (see section 4.5.3 for edge weight explanation) which is then utilised to find the final match strength between the user utterance and the database patterns. By adapting the edit distance algorithm to allow the flexibility of common spelling variations in the edit distance UMAIRs engine can calculate and assign more accurate edge weights to the tokenised words, therefore reducing the negative impact these group of commonly mistaken characters has on the final similarity calculation. The edit distance component of the WOW algorithm is now specifically tailored to address one of the language challenges unique to Urdu, making the similarity calculation more robust and accurate.

In order to further reduce impact of spelling mistakes on the PM/similarity engine the further techniques that could be adopted by UMAIR to help the users while typing/entering utterances to interact with UMAIR have been explored. This feature is outlined in the following section.

7.3 Predictive text

The predictive text feature was added to UMAIR based on research in to Human Computer Interaction (HCI) and methods of reducing spelling errors from the user perspective. Based on the literature it was found that in a text dialogue system a predictive text feature can aid users with spelling and reduce spelling errors while typing/entering utterances (Akram et al., 2014, Mora-Cortes et al., 2014, Kaufmann et al., 2012). Therefore, it was decided to implement a predictive text feature in to the architecture/UI of UMAIR to address the negative impact spelling errors have on the similarity calculation. The predictive text feature utilises an Urdu dictionary which contains 786 words created from the log file of the first UMAIR evaluation. The user utterances from the first evaluation were collated and validated for spelling errors once all the words were validated they were stored in the knowledge database as the dictionary for the predictive text component to utilise.

The predictive text feature is initiated when the user types the first letters of the intended word, all words from the attached dictionary that share the same first letters are activated, and the most frequently used word (see section 7.5 word frequency component) among them is presented to the user. The predictive text feature utilises the word frequency component in order to make intelligent suggestions to the user based on previous knowledge of user utterances. The suggested word is presented to the user highlighted in a lighter font colour within the input textbox. The user then can either further type the intended word which will then further narrow the list of activated words, or select and accept the predicted word as soon it appears in the textbox by pressing the left arrow key on the keyboard. An example is illustrated in Figure 36.

Figure 36- Predictive text feature from UMAIR UI

In the example above the user typing the Urdu word for ID card which starts with the Urdu letter ش, the predictive text system offers the suggested word شناختی based on the past frequency of this words usage with the system. One of the main causes of unrecognised utterances from the first evaluation stemmed from spelling related errors made by the user. The spelling related errors resulted in the engine failing to recognise that particular word when processing the user utterance. The predictive text feature is implemented in order to reduce the number of spelling related errors that occur during the user interaction by aiding the user while they are typing utterances in to the system. To date the work on Urdu predictive text is very limited and to the researchers knowledge this is the first predictive text system implemented on a non-mobile device.

7.4 Word segmentation algorithm/component

Inconsistent word segmentation is a language unique issue for the Urdu language. The magnitude of its impact on CA's was only brought to light through the end user evaluation of the first UMAIR prototype. As discussed in chapter 3 section 3.8, due to the morphological features of the Urdu language, the use of space to separate words by the users in certain cases during writing is entirely optional.

This feature of the Urdu language had severe detrimental effects for the PM/similarity engine of UMAIR, as the process of PM requires the tokenisation of the utterance in to its individual words which are then processed by the engine. The evaluation results found that during the input of text in cases where users had the option not to leave space (i.e. when the word ends in a non-joiner character), most users took advantage of this language feature and opted not to insert space between words.

An example of this is illustrated in Figure 37 where an example problematic utterance (translates to "I need a new ID Card") taken from the log file of the first evaluation is illustrated in both its forms (i.e. with and without consistent spacing) the green represents the use of white space to separate words.

Inconsistent use of white space	Consistent use of white space
مجھےنیاشناختی کارڈ چاہئے	مجھے نیا شناختی کارڈ چاہئے
Utterance Tokenisation result	
مجھےنیاشناختی کارڈ چاہئے	مجھے نیا شناختی کارڈ چاہئے

Figure 37 - Inconsistent and consistent word spacing

In cases such as the example illustrated in Figure 37 the engine tried to perform pattern matching on the whole token with all three words as one, which would cause the engine to fail to recognise that word/token thus negatively affecting the whole similarity calculation, and reducing the knowledge available in the utterance to the engine in relation to pattern matching. It was evident that this word segmentation issue had to be tackled in order to increase the effectiveness and robustness of UMAIRs engine, which relies on the user utterance to be correctly segmented in order to perform PM and similarity calculation more effectively.

Through research it was discovered that there were two possible options that could be adopted in order to mitigate this issue. Firstly, the scripts could be amended so that the scripted patterns included the inconsistently segmented versions of the patterns. The second option was to research and develop a new component that could insert spaces and segment, un-segmented/inconsistently spaced user utterances into valid words in real time before the utterance tokens were sent forward for processing by UMAIRs engine.

The first option although feasible was not the best option as this would further exacerbate the task of the scripter and involve further complexity during the scripting process. As all possible variations of the utterance with and without consistent segmentation would have to be scripted. In light of this a new Urdu word segmentation algorithm was researched, developed and implemented in to UMAIR's architecture which would pre-process the user utterances in order to ensure that the individual words of a particular user utterance were correctly/consistently segmented thus allowing UMAIR's engine to process the text without the hindrance of inconsistent word segmentation.

The general process the word segmentation algorithm follows in order to segment an utterance containing an unrecognised token follows is illustrated in Figure 38.

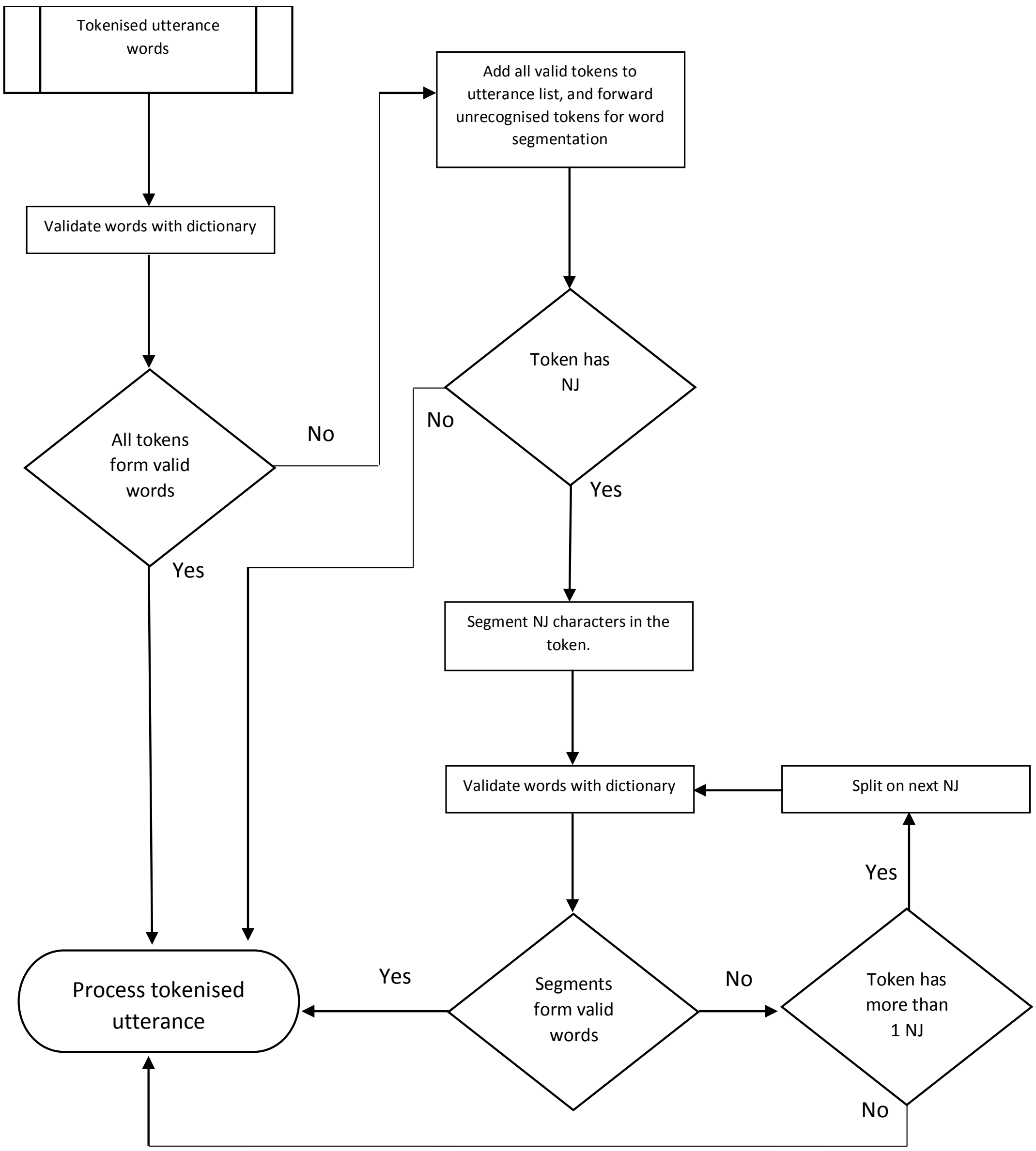


Figure 38 - Word segmentation process flow

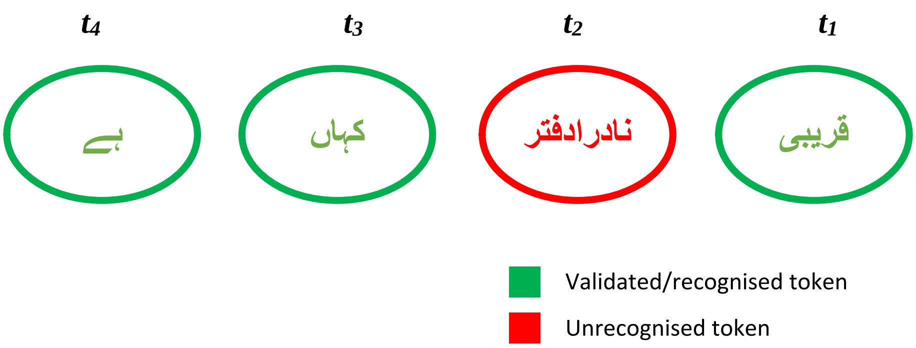
The word segmentation algorithm can be defined as follows: let the number of non-joiners be *nj*. *nj* is the total sum of the non-joiner characters in the token. The value of *nj* is used to measure the potential number of words (*npw*) in the token (illustrated in Equation 8). When the number of potential words (*npw*) in the token is calculated through Equation 8. The value of *npw* is the number of potential words that could be in the unrecognised token.

$$npw = \sum (nj + 1)$$

Equation 8 - Identify the potential words (npw) in token

To illustrate how the algorithm calculates the number of potential words in an utterance that contains an unrecognised token in the utterance consider the following example utterance:

Urdu: "قریبی نادرادفتر کہاں ہے؟"
Translation: "Where is the local NADRA office?"
This utterance will first be split on the white space in order to form tokens for the engine to process (i.e. perform pattern matching). Each split token is validated as a valid word through comparison to the Urdu dictionary (see section 7.8 for Urdu dictionary). The results of the tokenisation process for the example utterance is illustrated in Figure 39.



Unrecognised token will be processed by the segmentation algorithm with the intention of finding valid words within this token.

Figure 39 - Results of tokenisation

The number of non-joiner characters in t_2 are 5 (ادراد) (see Chapter 3 section 3.2 for Urdu morphology) thus according to Equation 8 the number of potential words (*npw*) in this token is 6 (*npw = 6*). The *npw* value is then utilised by the second part of the word segmentation algorithm that takes the unrecognised token (t_n) and splits that token on the non-joiner characters identified with it sequentially in order to find valid words from the Urdu dictionary (*ud*) illustrated in Figure 40. If the token when split on the non-joiners forms valid words that use all the characters in the token (i.e. remaining characters (*rc*) = 0) and the number of words formed are less than or equal to the number of potential words (*npw*) illustrated in Equation 9. Then these words are accepted and included in the utterance token list, which is then sent forward for processing by the engine.

$$if \quad \left((rc = 0) \; and \; \left(t_n \bigcap ud > 0 \right) and \; \left(t_n \bigcap ud \; \leq npw \right) \right)$$

Equation 9 – Validate split words from token

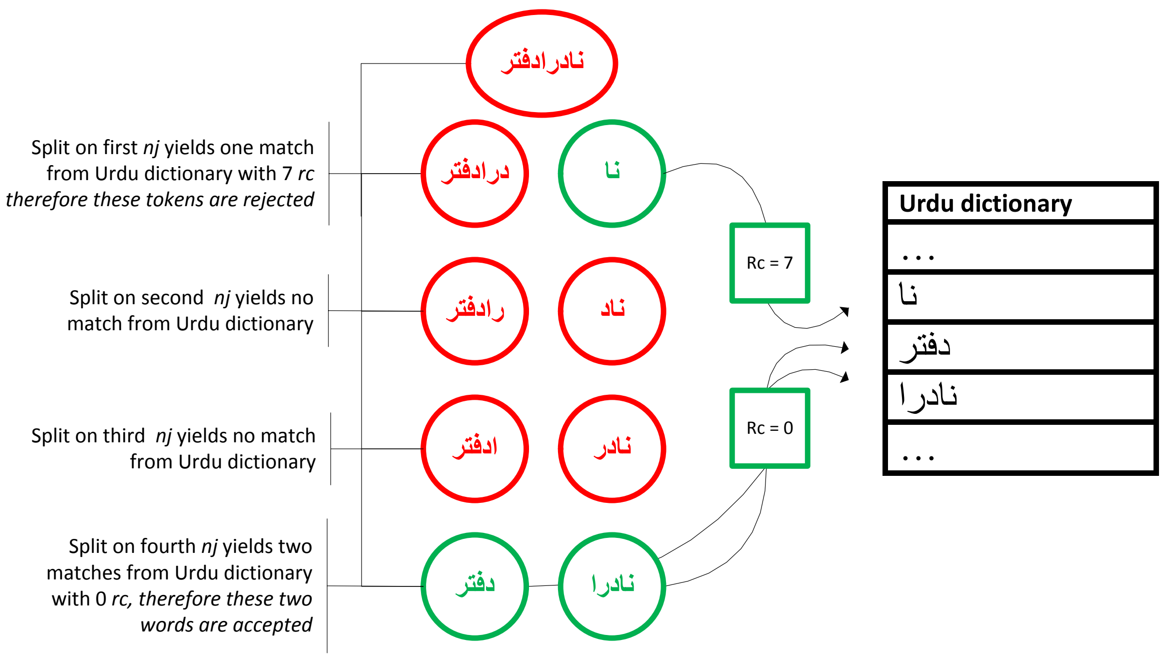


Figure 40 – Unrecognised token processing

When the split words of the token form valid words that are above 0 and = < the *npw* (Equation 9) the words found are added to the utterance and sent forward for the engine to process.

The word segmentation algorithm exploits the non-joiner (NJ) characters of the Urdu language which can be utilised to identify possible word segmentation boundaries. Nevertheless, the non-joiner characters are not a concrete indicator to word boundaries as they can appear in the middle of word. Word segmentation of utterances does pose some challenges such as over segmentation of words (Rashid and Latif, 2012). However, this has been reduced in this algorithm through the utilisation of the two Urdu dictionaries which are also used in the predictive text component (section 7.3). The first dictionary is a domain specific dictionary which is comprised of 786 domain specific frequently used words that were derived from the log file of the first evaluation (See section 7.8 for further details on the domain specific Urdu dictionary), the word frequencies in this dictionary are calculated by the word frequency component (see section 7.5 for further details on the word frequency component). The second dictionary is a general Urdu dictionary comprised of 2430 of the most frequently used ligatures which have been extracted from a 19.3 million Urdu word corpus gathered from a wide range of domains complied by the Centre for Language Engineering, Pakistan (Engineering, 2014). The domain specific dictionary contains the most frequently used words related to the domain of UMAIR making the dictionary smaller and more focused. The words in the domain specific dictionary take precedence over other general Urdu dictionary so the segmented words are first compared to this dictionary to identify the words, as the utterances are more likely to contain domain specific words which will reduce processing time. Furthermore, the domain specific words will take precedence over the general word dictionary in order to avoid over segmentation.

An example of over segmentation is demonstrated in word شناختی (identification), this word contains a non-joiner character within its ligature (ا). This word can be split to form the word شنا (define) that is also a valid word. However the use of the frequently used word dictionary mitigates this issue as words found within this dictionary take precedence over words found in the general Urdu dictionary, the word شناختی is found in the frequently used Urdu dictionary as it is a word that is frequently used in the domain of UMAIR. Moreover, for example if the word was not in the frequently used dictionary and the word شنا was found from the general dictionary, it would leave the remainder of the ligature which would be ختی this word has no meaning in Urdu, therefore the algorithm is programmed to reject both parts of the word (as the *rc* = 3) and continue processing through the unrecognised token until all segments form valid words which leave no remaining characters. Another step taken to avoid over segmented is the longer words found through segmentation take precedence over the shorter words. Thus in an instance described above where a word can be segmented and both parts of the word form valid words the algorithm is programmed to use the whole word not the two segmented words.

Once all the characters in the unrecognised token form valid words then these words are sent forward to be processed as valid tokens as a part of the original utterance to be processed by the PM engine. The pre-processing process of segmentation and validation ensures that non-segmented tokens are captured and processed, thus ensuring the only valid tokens are sent forwards to be processed which also maximises probability of finding a strong match to the utterance from the scripted patterns.

Table 13 illustrates some examples of how the algorithm pre-processes utterances in order to ensure consistent word segmentation so that all tokenised words from valid words. The example utterances are some of the utterances that the first prototype failed to recognise that were taken from the log file of the first UMAIR evaluation. The engine from the first UMAIR prototype failed to recognise these utterances because they contained instances where the user opted not to leave space after the non-joiner characters (highlighted in red). To a native Urdu reader there is no problem is distinguishing word boundaries, however for a PM engine that relies on the use of consistent white space to tokenise words this caused major problems.

UMAIR 1 **Without word segmentation component**	**UMAIR 2** **With word segmentation component**
قریبی نادرادفتر کہاں ہے؟ Where is the local nadra office?	قریبی نادرا_دفتر جہاں ہے؟

کونسافارم بھرناہوگانیاپاسپورٹ کےلئے؟ Which form do I have to fill in for a new passport?	کونسا_فارم بھرنا_ہوگا_نیا_پاسپورٹ کے_لئے؟
	Continued…

ایک نئےپاسپورٹ کتناکا ہے؟ How much is a new passport?	ایک نئے پاسپورٹ کتنا کا ہے؟
میں پاکستان سےنہیں ہوں I am not from Pakistan	میں پاکستان سے نہیں ہوں

Table 13 – Utterances before and after being processed by word segmentation algorithm

Table 13 shows how the word segmentation algorithm processed the inconsistently segmented user utterances to ensure all the tokenised words formed valid words. The green space in the UMAIR 2 column highlight where the algorithm segmented the tokens to from valid words.

7.5 Word frequency component

Word frequencies are used in many widely used practical applications of statistical natural language processing, such as document retrieval based on keywords (Altmann et al., 2009). The word frequency component was added to UMAIR's architecture in order for UMAIR to be able to learn and adjust word frequency values in the domain specific dictionary according to the data stored in the log file. The word frequencies work with the word segmentation and predictive text components in order to offer intelligent and more relevant suggestions for both these components. These components both utilise dictionaries in order to mitigate Urdu language specific issues. However by calculating word frequencies both these components are able to operate more intelligently and effectively, by offering more appropriate suggestions to the predictive text component (see section 7.3) based on the frequencies of words used in previous utterances. Furthermore, the word frequencies are utilised by the word segmentation component/algorithm (see section 7.4) to resolve over segmentation and instances where tokens can be segmented in multiple variations, in these instances the words with the higher frequencies take precedence over the less frequently used words.

The original word frequency value was calculated and stored in the database through the knowledge captured and stored in the log file. The log file stores anonymous data of all the user utterances that are processed by UMAIR's engine. The values are also updated at the end of each discussion where the log file records of the conversation are automatically scanned and all valid words used by the user during the conversation are captured and used to update the frequency values stored in the database. This data/knowledge is then utilised by the word frequency component to calculate and adjust the word frequency dictionary, in order to offer intelligent suggestions through the predictive text feature and to improve the word segmentation algorithm.

The word frequency component utilises the Bags of Words (BOW) (Boulis and Ostendorf, 2005) technique to calculate the word frequency (see Equation 10 – Bag of Words Frequency Equation). The bag-of-words retrieval models represent queries and documents as unordered sets of terms; this strategy is based on an independence assumption. Bag-of-words models have been shown to be simple and effective (Choi et al., 2014). The bag-of-words representation, is represented with a vector of the word counts that appear in it. Depending on the classification method, the bag-of-words vector can be normalized and scaled (Boulis and Ostendorf, 2005).

The ranking functions associated with bag of words retrieval models often consist of term frequency (Metzler, 2008). In addition to using words as indexing terms it is usually assumed that the ordering of the words does not matter in this instance as this implementation is only concerned with calculating word frequencies, not word or sentiment classification. This way utterances no longer have to be represented as sequences. Instead the utterances can be represented as a bag of words. This representation is equivalent to an attribute-value representation as used in machine learning. Each distinct word is a feature and the number of times the word occurs in the log file/temporal memory is its value. This is represented by the following equation:

$$TF\ (w, d)$$

Equation 10 – Bag of Words Frequency Equation

TF value is called the term frequency, thus, TF equals (w, d) of word w in a document/log d. The calculated TF values are stored as a variable in the knowledge base, which is then utilised by the engine during word suggestion and segmentation to improve both processes.

7.6 Knowledge base expansion

The knowledge base was expanded in order to address the finding from the end user questionnaire which was administered to the participants of the first end user evaluation. The participants expressed that they perceived their discussion to be low in naturalness, meaning the participants thought that their dialog with UMAIR was "robotic". In order to address this finding the knowledge base was expanded to make UMAIR more natural in terms of conversation.

This was done by firstly increasing the domain specific knowledge implemented in UMAIRs knowledge base. In order to expand the knowledge, the domain was knowledge engineered with the intention of adding new contexts to the knowledge base (see chapter 4 section 4.6.2 for knowledge engineering the domain). This process involved further interviews with the NADRA industry contact in order to understand the business logic involved with the additional knowledge to be included into the knowledge base (see appendix J for interview question). The knowledge base was expanded to include knowledge on passport application as well as ID card application. All the unrecognised utterances resulting from weakness in the knowledge base from the first evaluation were added as new patterns to the knowledge base.

Furthermore, the FAQ layer of the knowledge base was expanded to include more FAQ with relation to the domain. Lastly, more responses were implemented in to UMIARs knowledge base. The structure of the knowledge base was amended in order to allow more responses for each rule to be scripted. This varied the responses deliver to the users simulating more variety in the discussion, and making UMAIR less repetitive.

7.7 Short term memory

A short term memory feature has been added to UMAIRs architecture to address the naturalness of the discussion. In order to communicate through dialog some form of memory is essential. Human memories can be triggered through the use of clues, cue words and through the use of semantic relations. To simulate a more intelligent, human-like dialogue, CA's design must incorporate an aspect of memory (O'Shea, 2011). Memory in virtual agents has typically been implemented to address the issue of how agents remember information from one interaction to another to simulate a more natural human like conversation. This is considered necessary for agents to effectively carry out the role for which they are designed (Richards and Bransky, 2014). Brom and Lukavský (2009) have also stated and emphasised the need for memory in conversational agents. They state that it is necessary for agents to utilise memory for a broad range of tasks like debriefing, giving information, remembering the course of interactions, searching for objects, knowledge sharing and learning; noting that the important concept behind intelligent virtual agents is believability, where the primary goal is to produce agents that imitate human like behaviour.

It was discovered through the analysis of the subjective evaluation data from the first prototype that the general consensus from the participants with relation to the naturalness of conversation was low. The feedback received from the majority of the participants expressed that the conversation was not natural, repetitive and robotic. In order to address this a short term memory feature was researched and developed and included in UMAIR's architecture. This feature allowed UMAIR to remember the rules that were fired during each individual conversation, this meant that if a user was to repeat a question UMAIR was able to respond with a different answer and also say to the user "as we discussed earlier…" (Or a set variation of this phrase). This made the conversation more natural and intelligent and less repetitive, as UMAIR could simulate a short term memory. Hence when a user repeated an utterance or the same rule was fired twice during the conversation, UMAIR was able to respond more intelligently with different responses for each repetition depending on the context.

Additional variables were added to the scripting language to allow multiple response to be scripted for each rule, each response is tailored depending on the amount of repetition that had to be made by UMAIR by utilising the short term memory. An example of this is illustrated in Table 14.

Context – N/A
Rule – no_match Response 1: sorry I didn't understand Response 2: I still do not understand what you are trying to say, could you please try using different words Response 3: Sorry I don't think it's possible for me to help you with this matter, I suggest you speak to one of our representatives in person by visiting your local NADRA office Switch Context: null Switch to: null Support material: poc_form.pdf Requires Vars: No Allow Yes/No Tree Node Pos: null Tree Node Neg: null Max Repetition: 3

Table 14 – Example of updated scripting language

The example rule in Table 14 is the rule which is fired when a user utterance was not recognised. The first UMAIR prototype simply responded by saying "sorry I didn't understand you" each and every time the failed to find a match for the utterance, if the utterance was consecutively not understood by UMAIR the same response was delivered indefinitely. However in order to make UMAIR seem more intelligent and less repetitive, the scripting language now allowed a maximum number of repetition, which each repetition the response is different, furthermore depending on the situation if the rule is repeated more than the allowed number of repetitions (variable 'Max Repetitions' in the scripting language) the conversation is terminated. An example of this is outline in Table 15.

Short term memory – unrecognised utterance rule	
Iteration	**UMAIR Response**
1^{st}	"sorry I didn't understand"
2^{nd}	"I still do not understand what you are trying to say, could you please try using different words"
Final	"Sorry I don't think it's possible for me to help you with this matter, I suggest you speak to one of our representatives in person by visiting your local NADRA office"

Table 15 - example of responses based on short term memory

In the example shown in Table 15, once the final response is delivered to the user the conversation is terminated, however if the user changes the utterance to something UMAIR is able to recognise, then the conversation is continued. This allows UMAIR to respond to unrecognised and repeated utterances in a more intelligent manner.

The short term memory component was added to the conversation manger (see Chapter 4 Section 4.2.1.1 for conversation manager). The conversation manager utilises the short term memory to be able to manage the conversation more intelligently and be able to mitigate the problem of the user just repeating the unrecognised utterances which caused frustration which eventually lead to the users giving up the interaction. The addition of short term memory in to the architecture allows UMAIR to respond to user with more intelligence by utilising previous knowledge related to the discussion thus making the conversation and responses more natural and seemingly aware of previous discourse.

7.8 Urdu Domain Specific and General Dictionary

Two Urdu dictionaries have been added to the knowledge base of the system which are utilised by several architecture components detailed in the preceding sections. The first dictionary is a domain specific dictionary which is comprised of 786 domain specific frequently used words that were derived from the log file of the first evaluation. The log files contained records of all the participant conversations from the first evaluation, these conversations were scanned and all unique words found were validated and added to the domain specific dictionary.

The domain specific dictionary contains the most frequently used words related to the domain of UMAIR making the dictionary smaller and more focused. The words in the domain specific dictionary take precedence over other general Urdu dictionary so the segmented words are first compared to this dictionary to identify the words, as the utterances are more likely to contain domain specific words which will reduce processing time.

The second dictionary is general Urdu dictionary comprised of 2430 of the most frequently used ligatures which have been extracted from a 19.3 million Urdu word corpus gathered from a wide range of domains complied by the Centre for Language Engineering, Pakistan (Engineering, 2014).

7.9 Improved/Updated User Interface

The UI of the second UMAIR prototype was improved by adopting some of the embodiment techniques discussed in Chapter 2 Section 2.4. Embodied conversational agents are computer-generated characters that demonstrate many of the same properties as humans in face-to-face conversation, including the ability to produce and respond to verbal and non-verbal communication (Cassell, 2000a, Derrick and Ligon, 2014).

From the evaluation of the first UMAIR prototype it was found that the user's perception relating to the UI was quite negative and some of the comments expressed its lack of engagement and plain design (see chapter 4 Section 4.5.13 for prototype one UI). The UI of the second prototype was adapted to include an animated character, in order to make it more engaging and natural for the users. The updated UI is illustrated in Figure 41. Embodied characters have been used to provide feedback and visual stimulus for users during their discussions in many CA's (Tegos et al., 2014, Nunamaker et al., 2011) .

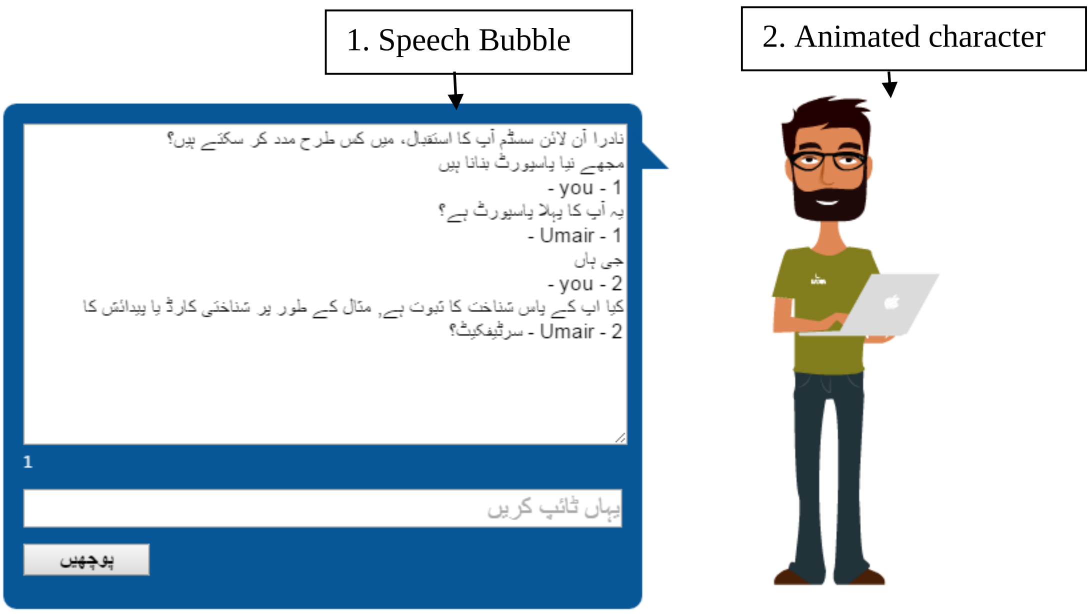
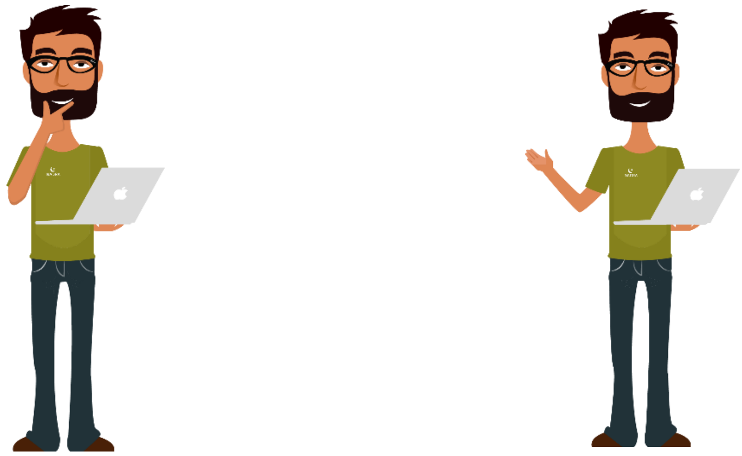


Figure 41- Updated UMAIR UI

The textboxes where the user entered text and received responses from UMAIR has been adapted to look similar to a speech bubble (1). The intention behind this was to show that the conversation/response is from the embodied character to simulate naturalness and a connection between the response and the charatcer. The character (2) included in the UI is a visual representation of UMAIR. When the user first accesses the system the character introduces itself as UMAIR and asks the user “how can I help you?” in order to initialise the conversation. The UMAIR character also provides visual clues such as pointing and other gestures to give visual aides to the user where necessary depending on the context of the discussion (illustrated in Figure 42).



Didn’t understand/thinking	Pointing to interface object (i.e. textbox document, picture etc.)

Figure 42 – Embodied character actions

7.10 Updated Architecture Diagram

Figure 43 illustrates the updated architecture of UMAIR. The figure outlines all the new components presented in this chapter and how they interact with each other to overcome the issues highlighted through the first end user evaluation.

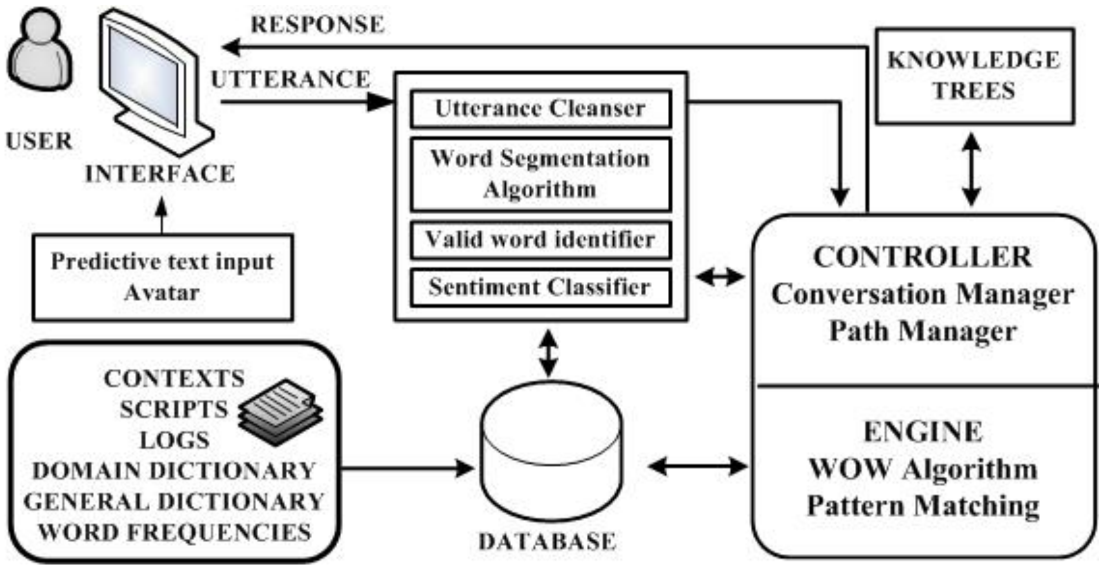


Figure 43 – Updated Architecture

7.11 Scripting tool

A scripting tool (illustrated in Figure 44) has been developed with the aim of making the task of scripting easier. The scripting tool is a not connected to the engine its sole purpose is to aid the scripter when scripting patterns to be stored in the knowledge base. The tool acts an interface between the scripter and the knowledge base/database.

Figure 44 - Scripting tool UI

The tool allows the scripter to add, edit and delete patterns, rules and responses with ease. The researcher found that scripting by directly interacting with the database was a cumbersome and error prone process. The scripting tool allows the scripter to easily enter and amend patterns and rules in the database while being able to set all the variables (i.e. decision tree interaction, allow word order, allow yes no, supporting media etc.) in the scripting language through the scripting tool GUI.

7.12 Chapter summary

This chapter has outlined the additional research, development and approaches undertaken to address the weaknesses brought to light during the evaluation of the first UMAIR prototype. The components developed at this stage of the research further enhance and bolster the effectiveness and robustness of UMAIR. New word segmentation and predictive text features backed by the word frequency component have been added to the architecture in order to improve the robustness and accuracy of UMAIRs engine. Additional supplementary components such as a short term memory, an improved UI, predictive text feature and further refinement of the WOW algorithm have been researched and implemented with the intention of improving the overall effectiveness and user experience of UMAIR. The new updated architecture of UMAIR will undergo end user evaluation with the intention of gauging whether or not the new components have any impact on the success and effectiveness of UMAIR as a CA compared to the first prototype. The evaluation methodology and results are outlined in the following chapter.

Chapter 8 - UMAIR Phase Two Evaluation Methodology & Results

8.1 Introduction

The first phase of evaluation was aimed at validating the Urdu CA Framework methodology and the implemented Urdu CA UMAIR. During the first phase of evaluation there were a number of lessons learned along with areas for improvement and further development were highlighted. In order to address these issues they were individually researched which subsequently lead to improvements and additions to several feature of UMAIRS architecture.

Phase one of UMAIR’s evaluation focused on evaluating metrics related to the different components of UMAIR’s architecture. These metrics will be carried over into the second phase in order to gauge the success of the enhancements made to the components in UMAIR’s architecture and overall competence as an effective Urdu Conversational Agent. Each of the metrics map to different features of UMAIRs architecture, therefore the metrics can be used to perform analysis on each aspect individually in order to measure its success and contribution to the overall architecture. These metrics will form the benchmark which the metrics from the second prototype system will be compared. The intention behind this is to bring to light any significant improvements between the metrics in the two systems.

8.2 Experiment Design

Data for this phase of testing was gathered through experiments which consisted of end user evaluation where participants interact with the system and subsequently fill out a user satisfaction/usability questionnaire. The aim of the evaluation is primarily to measure the success of developed components from phase 2 of the research and the impact they have on the overall effectiveness of UMAIRs engine. This will bring to light whether the developments made improve and enhance UMAIRs conversational abilities while tackling the problems that where highlighted during the first phase testing/evaluation. The results derived from this stage will contribute towards concluding the main research question.

8.3 Hypothesis

The original objective of this research is outlined through the main research question which is as follows:

Research Question: – It is possible to produce an effective Urdu CA.

The following main (**H1**) and subsidiary hypothesis are to be tested through this second evaluation in order to test UMAIRs updated architecture. The hypothesis correspond to the objective and subjective aspects of the second UMAIR prototype. In each case the null hypothesis is no effect, if there is an effect, the experimental value may indicate improvement or deterioration in the tested component.

H1- the enhancement made to UMAIRs architecture improve the overall effectiveness and robustness of UMAIRs engine.

> **H1-A.** The improvements and changes made to the WOW algorithm have an impact on the accuracy and effectiveness of UMAIRs engine and reduces the percentage of unrecognised utterances.
>
> **H1-B.** The addition of the word segmentation feature made an impact in improving UMAIRS engine in terms of reducing the rate of unrecognised utterances.
>
> **H1-C.** The addition of the predictive text feature made an impact in improving UMAIRs engine by reducing the rate of unrecognised utterances.
>
> **H1-D.** The improvements made to UMAIRs result in better perceptions from the users in relation to the subjective metrics.

The main hypothesis (H1) will be accepted or rejected based on the results of the subsidiary hypothesis (A, B C and D).

8.4 Experiment

Evidence for H1 is gathered through a between groups experiment, in which the data gathered from this evaluation group will be compared to the data gathered from the evaluation group of the first prototype. The second prototype is hoped to perform significantly better in terms of objective task completion criteria. The end user experiments consists of the users interacting and conversing with UMAIR in order to solve a query or problem relating to the domain. The interaction between the users and UMAIR will produce log files which will record the objective metrics related to the discussion between the participant and UMAIR. These logs will be utilised to measure the success of the newly researched and developed components added to UMAIRs architecture in the second phase.

The subjective metrics will be captured through the end user questionnaire (see Appendix B for questionnaire of second analysis), which is adapted by adding more questions to measure the users perceptions with regards to the enhancements made to the architecture for example their opinion about the predictive text feature. The combination of the log files and end user questionnaire will generate the objective and subjective data that can be collated and statistically analysed in order to gauge the effectiveness of the enhanced architecture. This is illustrated in Table 16.

	UMAIR Prototype 1	**UMAIR Prototype 2**
Objective	Log file results of original components and architecture (Old participants)	Log file results of enhanced and additional components (New participants)
Subjective	Questionnaire results (Old participants)	Updated questionnaire results (New participants)

Table 16 - System evaluation methods

8.5 Participant interaction

The system was deployed online and participants were invited through an email link to take part in the experiment. The participants were briefed via the email that the system is a prototype and that it can only answer questions related to the domain of NADRA specifically ID card, and passport application. They were told that the scenarios are only guidelines to specify the possible tasks that the agent could address and that they were free to go ahead and interact with the system as they felt appropriate (e.g. language used) in order to complete the scenario based task assigned to them.

The participants were given their particular problem/scenario related to the domain prior to them using the system, and were instructed to ask UMAIR how to solve their particular problem (see section 8.9 for scenarios).

The participants selected were fluent in both Urdu and English. There are two reasons behind this, firstly because the participant will interact with UMAIR in Urdu and fill out the questionnaire in English. Secondly, the participant will receive instructions with regards to their scenario in English, and will then interact with the system in Urdu. This design choice has been made intentionally as not to introduce a bias in the language the participants used through the instruction they received in their particular scenario.

8.6 Evaluation Metrics

The metrics in **Table 17** and **Table 18** were derived using the GQM methodology which was utilised in the first phase of UMAIRs evaluation (chapter 2 section 2.9). These metrics will be compared to the metrics from the first set of data to see if there is any statistically significant improvement between the two data sets.

SUBJECTIVE METRICS		
Metric to be Evaluated	**Mode of Evaluation**	**Characteristic Measured**
Agent naturalness	Questionnaire	Usability/user satisfaction
User Interface (UI) design	Questionnaire	Effectiveness of the UI/user satisfaction
Time take to get information required	Questionnaire/Log File	Usability/functionality
Overall user satisfaction	Questionnaire	Overall effectiveness of the UCA from end users perspective

Table 17 - Subjective evaluation metrics

OBJECTIVE METRICS		
Metric to be Evaluated	**Mode of Evaluation**	**Characteristic Measured**
Number of correct responses	Log file	Agent accuracy/robustness
Number of Incorrect responses	Log file	Agent accuracy/robustness
Number of unrecognised utterances	Log file	Agent robustness/robustness
Agents ability to understand user utterances	Log file	Agent robustness/robustness
Number of utterances requiring word segmentation	Log file	Ability to segment words in order to increase robustness and accuracy
WOW algorithm processed utterances	Log File	Algorithms ability to handle word order variation Effectiveness of the similarity calculation
Number unrecognised utterances	Log File	Scripting/robustness
Goal of conversation achieved	Log File	Agent effectiveness/robustness

Table 18 – Objective evaluation metrics

8.7 Data Collection

8.7.1 Subjective Data Collection

The data to test the subjective measures will be gathered through an end user questionnaire. The questionnaire is updated to include questions that address the research questions at this stage in the research.

8.7.2 Objective Data Collection

The data to measure the objective measures will be derived from the log file generated from the participant’s interaction with UMAIR. The log file records discourse related metrics about the user’s discussions with UMAIR.

8.8 Data Analysis

The data gathered will be statistically analysed and compared to the data gathered from the previous prototype of UMAIR. This will highlight which if any of the new additions, enhancements and developments to the components in UMAIR's architecture have any significant impact on the effectiveness of the engine. This will highlight the extent of the impact the individual additions, enhancements and developments have on the engine. The selection and application of statistical analysis techniques will be determined subsequent to the evaluation and will be directed at answering the research hypothesis. This will entail some between group's analyses, to highlight the differences in the data between the two tested prototypes of UMAIR. These results will be utilised to test the research hypothesis.

Moreover as the second prototype is tested in Pakistan as well as the UK, the data analysis also includes a comparison between the data gathered from the two countries in order to shed light on the differences, if any, between the datasets gathered from the two countries. This brings to light whether or not the participant's location and first language makes any difference in the way they interact with the system and whether or not these variables have an impact on the effectiveness of UMAIRs. The data analysis groups are illustrated in Figure 45.

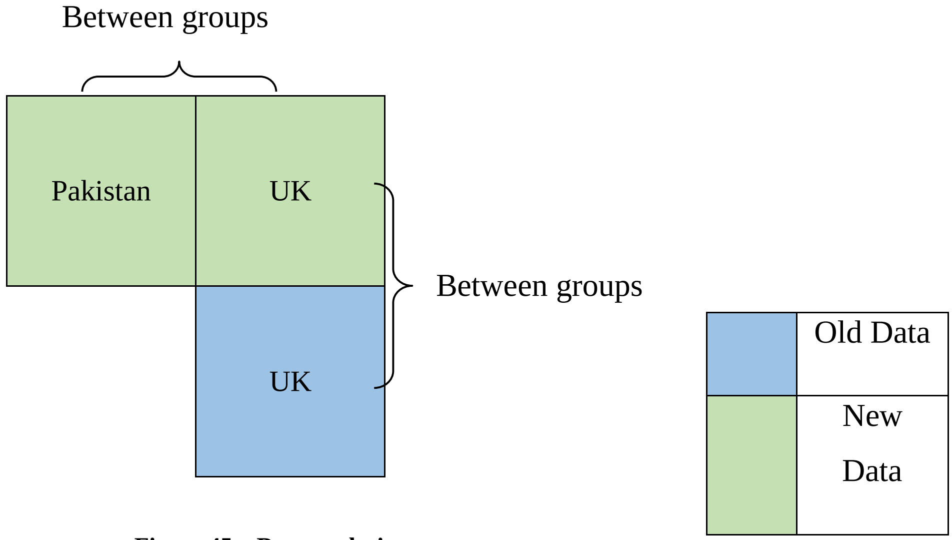


Figure 45 – Data analysis groups

8.9 Scenarios

Scenario-based evaluation methods evaluate software's ability with respect to a set of scenarios of interest which are based on the goals/objectives of the developed software. A scenario is a brief descriptions of a single interaction of a stakeholder/participant with a system (Roy and Graham, 2008). For this stage of evaluation the knowledge base has been expanded through further knowledge engineering (chapter 7 section 7.6) so the scenarios are increased to include ID card and passport application. The scenarios are all based on real world queries collected through the knowledge engineering stage, and are all scenarios that the NADRA department receive and deal with on a daily basis. The scenarios are a mixture of complex and simple tasks related to the domain which have been devised and validate through the industry contact at NADRA. The participants are tasked to complete either one complex or two simple scenarios as a part of their interaction with UMAIR.

8.10 Sample

The size of the sample was significantly increased (70 participants) for this phase of the evaluation in order to gather more data, which will result in more conclusive and decisive results. Moreover the sample will be categorised into groups in order to analyse if participant location (UK and Pakistan) has any impact of the effectiveness of UMAIR. In order to give the agent a more thorough testing during this evaluation the system was also tested in Pakistan which is the main targeted demographic for the system. This will give a broader perspective in the data captured and further analysis can be made to highlight any differences between the UK sample and the Pakistan sample in terms of the effectiveness of UMAIR. The following section provides a descriptive analysis of the participant's sample.

8.10.1 Sample distribution by location

Figure 46 illustrates the frequency distribution of participants by location.

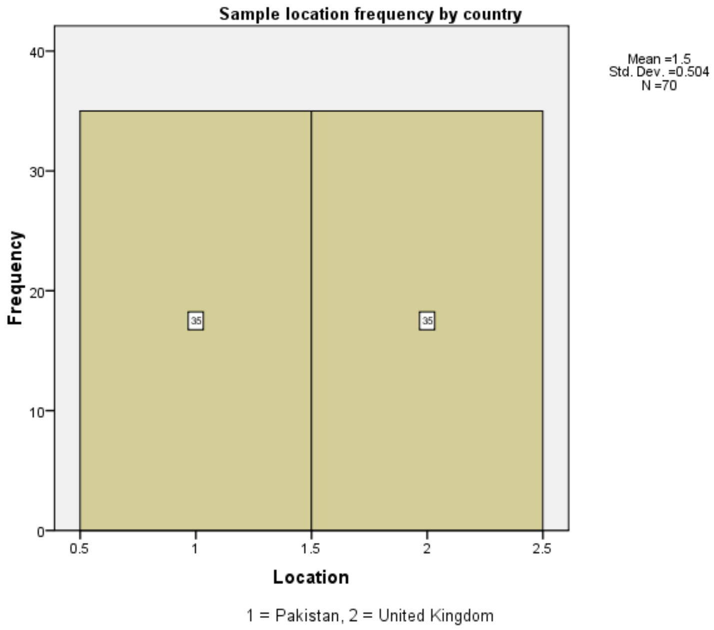


Figure 46 - Histogram of sample location

It can be seen from Figure 46 that the participants involved in this evaluation are evenly distributed through both locations (35 Pakistan, 35 United Kingdom). This distribution was intentional, as the author sought to gather data from both locations in order to highlight whether or not there were any differences in the way the users interact with UMAIR in the different locations. Furthermore, only data that was deemed to be complete data was kept for analysis, some data was omitted from the final datasets because some of the participants did not finish the full experiment. Thus, only participants who completed the full experiment (i.e. complete discussion with UMAIR and filled out questionnaire) were included in the analysis.

8.10.2 Sample distribution by gender

Figure 47 shows the frequency of the sample by gender.

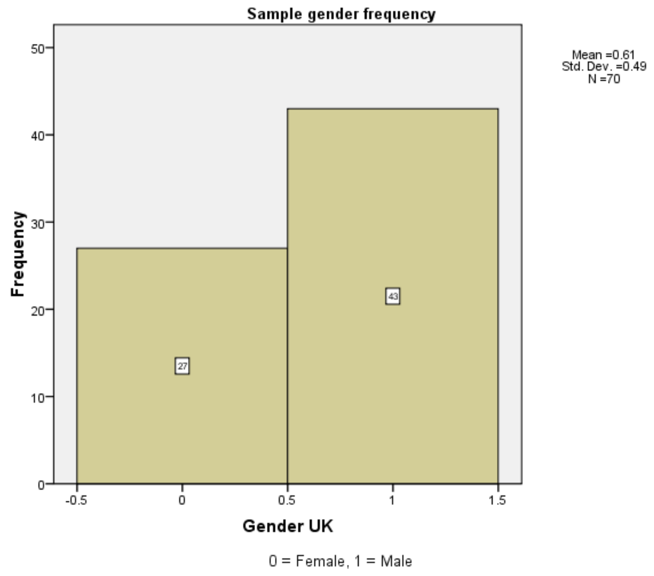


Figure 47 – Histogram of sample gender

It can be seen that the evaluation sample contained more male than female participants (43 male, 27 female). The distribution of gender is not exactly equal this is due to the fact that a convenience sampling methodology was adopted by the author. However, the sample has a good representation of both genders.

8.10.3 Sample distribution by age

Figure 48 illustrates the frequency of sample by age.

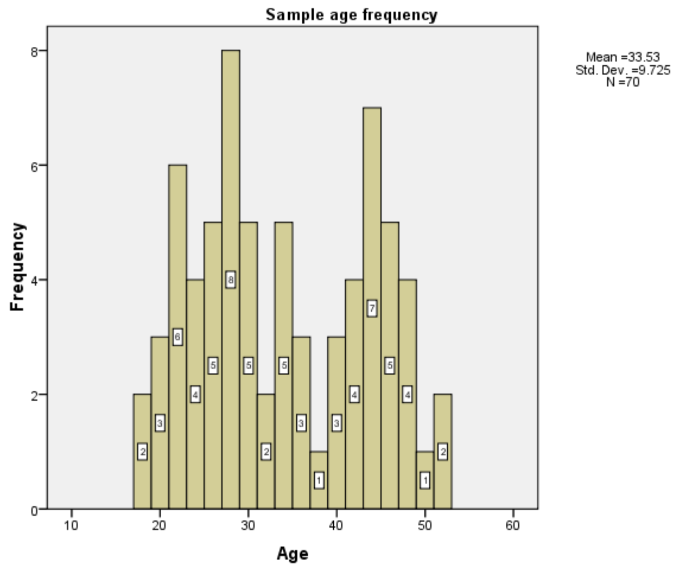


Figure 48 - Histogram of sample age

It can be seen that there is a wide age range represented within the evaluation participant sample (age 18 through 54). The age was considered by the author in order to highlight the differences, if any, between the different age groups and their interaction with UMAIR.

8.10.4 Sample distribution by education level

Figure 49 illustrates the education level of the total sample distribution in a pie chart and Figure 50 illustrates the education level of the sample between the two locations.

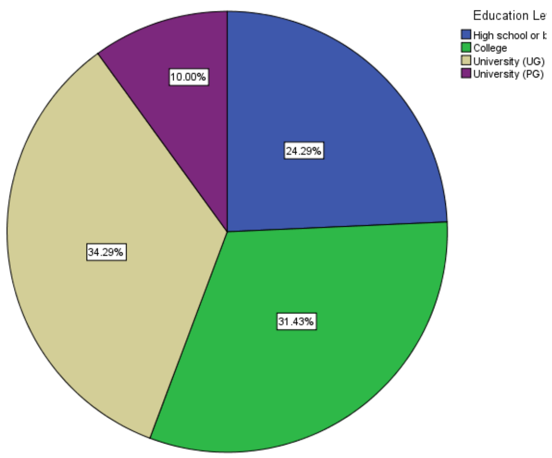


Figure 49 - Pie chart of sample education level

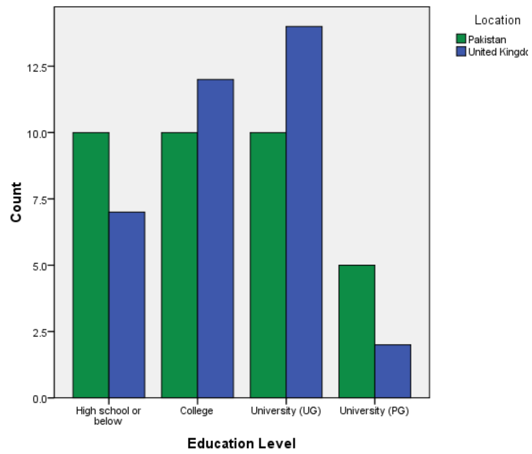


Figure 50 – Bar graph showing education levels between sample locations

It can be seen from Figure 49 that the majority (34.29%) of the participants from the sample were educated up to university undergraduate level. This was followed by 31.43% who were educated to college level, while 24.29% classified themselves as being educated up to the level of high school or below and finally 10% of the sample was made up of participants who classified their education level as postgraduate. The education level of the participant sample was collated in order to highlight differences, if any, between the education level groups and their interaction with UMAIR.

8.11 Results

This chapter presents the results from the evaluation of the second UMAIR prototype. The data analysis in the following sections is split in to four parts. The first section provides an overview of the results related to the objective metrics of the participant's conversation with UMAIR, which was gathered through the log file. The second section, consists of descriptive statistical analysis of the objective/log file data. The data from the second prototype is compared to the data gathered from the first prototype, in order to highlight any significant changes in the systems performance. The third section delves in to further statistical analysis of the data, in order to shed light on any differences in between the data gathered from the two different countries (UK and Pakistan) during the evaluation of the second UMAIR prototype. Finally section four presents the results of the questionnaire data which was employed to gauge the participant's perceptions with regards to the subjective metrics related to their interaction with UMAIR.

8.12 Log file analysis

The raw data from UMAIR's log file was collated, processed and analysed. Table 19 summarises the results of data gathered through the log file during the evaluation of the second prototype.

Log File Analysis (objective metrics)	
Total number of conversations	70
Total number of conversations UK	35
Total number of conversations Pakistan	35
Total number of conversations leading to goal achievement	68
Total number of utterances in all conversations	537
Total number of unrecognised utterances	17
Total number of WOW processed utterances	156
Total number of utterances requiring word segmentation	32
Percentage of unrecognised utterances	3.17 %
Percentage of WOW processed utterances	29.05 %
Percentage of utterances requiring word segmentation	5.96 %
Percentage of conversations leading to goal achievement	97.14 %
Average time per discussion (mins)	2.4

Table 19 – Log file analysis

The results presented in Table 19 illustrate that the second prototype performed well during the end user evaluation. These results are further analysed in the following section using statistical analyses techniques in order to determine whether these results are significantly different from the results of the first prototype.

8.13 Descriptive analysis of objective data from old and new data sets

This section presents tests performed to explore the differences and/or relationships in the data in order to test the subsidiary hypotheses (A-D) which will aid in concluding the main hypothesis H1 as discussed in section 8.3. Mann-Whitney U Tests are used to test the significance of the results.

8.14 Mann-Whitney U test

This technique is used to test for differences between two independent groups on a continuous measure. Mann-Whitney U is the non-parametric alternative to the t-test for independent samples. Instead of comparing means of the two groups, as in the case of the t-test, the Mann-Whitney U test actually compares medians. It converts the scores on the continuous variable to ranks, across the two groups. It then evaluates whether the ranks for the two groups differ significantly. As the scores are converted to ranks, the actual distribution of the scores does not matter (Pallant, 2004).

The following section presents the descriptive analysis of the objective data from the end user evaluation gathered through the log file. The analysis of results presented in this part are directly related to the answering the research hypothesis A, B and C. The findings are as follows:

8.14.1 Number of unrecognised utterances

Table 20 shows the results of the Mann-Whitney U test carried out to determine if there was any difference in number of unrecognised utterances between the first and second prototypes.

Ranks

	Dataset	N	Mean Rank
Number of unrecognised utterances	Prototype 1	24	63.17
	Prototype 2	70	42.13
	Total	94	

Test Statistics[a]

	Number of unrecognised utterances
Mann-Whitney U	464.000
Wilcoxon W	2949.000
Z	-4.206
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: Dataset

Table 20 - Mann-Whitney U test for unrecognised utterances

As Table 20 shows p-values less than 0.05 (p = .000), it can be concluded that there is a statistically significant difference in the number of unrecognised utterances between the two prototype systems. When comparing the mean ranks, of the two systems the first prototype was ranked higher, which highlighted the first prototype as having more unrecognised utterances.

8.14.2 Time taken to reach the goal of the conversation

Table 21 illustrates the results of the Mann-Whitney U test conducted in order to measure whether there was a statistically significant difference between the times taken to reach the goal of the conversation between the two prototype systems.

Ranks

	Dataset	N	Mean Rank
Total Duration (mins)	Prototype 1	24	50.13
	Prototype 2	70	46.60
	Total	94	

Test Statistics[a]

	Total Duration (mins)
Mann-Whitney U	777.000
Wilcoxon W	3262.000
Z	-.562
Asymp. Sig. (2-tailed)	.574

a. Grouping Variable: Dataset

Table 21 - Mann-Whitney U test for time taken to reach goal

From this result, it can be concluded that the time taken to reach the conversation goal between the two prototypes systems was not statistically significant ($p = .574$). In comparing the mean rank there is a nominal difference between them the first prototype system ranks higher, which indicates it took marginally longer for the users to reach the goal on the first prototype of UMAIR.

8.14.3 Number of utterances processed by WOW algorithm

Table 22 illustrates the results of a Mann-Whitney U test carried out to highlight if there was any significant difference between the number of utterances needing processing by the WOW algorithm (i.e. how many of the utterances in all conversations were unscripted word order variations of scripted patterns).

Ranks

	Dataset	N	Mean Rank
Number of utterances with WOW	Prototype 1	24	40.33
	Prototype 2	70	49.96
	Total	94	

Test Statistics[a]

	Number of utterances with WOW
Mann-Whitney U	668.000
Wilcoxon W	968.000
Z	-1.521
Asymp. Sig. (2-tailed)	.128

a. Grouping Variable: Dataset

Table 22 - Mann-Whitney U test for number of utterances needing WOW processing

The results illustrate that there is not a statistically significant ($p = .128$) difference between the number of user utterances which were word order variations of scripted patterns between the two prototype systems.

8.14.4 Goal Achievement

Table 23 outlines the results of a Mann-Whitney U test conducted to test if there was a statistically significant difference between the rates of conversation goal achievement between the two prototype systems.

Ranks

	Dataset	N	Mean Rank
Goal achieved	Prototype 1	24	41.21
	Prototype 2	70	49.66
	Total	94	

Test Statistics[a]

	Goal achieved
Mann-Whitney U	689.000
Wilcoxon W	989.000
Z	-2.879
Asymp. Sig. (2-tailed)	.004

a. Grouping Variable: Dataset

Table 23 - Mann-Whitney U test for goal achievement

From the results it can be deduced that there is a statistically significant difference between the two prototype systems ($p = .004$). When comparing the mean ranks it can be seen that the prototype 2 ranks higher than prototype 1, meaning that the goal achievement for prototype 2 was significantly improved compared to the goal achievement of prototype 1.

This concludes the results analysis of the log file data. The next section aims to further explore the collated data in order to highlight any differences that may be present between the data collated from the two evaluation locations (United Kingdom and Pakistan). This will provide further insights in to whether or not the participants from the two locations demonstrated any significant differences during their interaction with UMAIR during the evaluation.

8.15 Comparative descriptive analysis of data between locations

This section presents further descriptive analysis of the quantifiable data from the end user evaluation gathered through the log file. The analysis of results presented in this part are intended to highlight differences in the results datasets gathered from Pakistan and the UK. The Man-Whitney U test was employed in order to highlight any statistically significant differences in the data. The findings are as follows:

8.15.1 Duration of conversation between the locations datasets

Table 24 illustrates the results of a Mann-Whitney U test conducted to investigate whether there was a difference in the time taken to reach the goal of the discussion between the two evaluation locations.

Ranks

	Location	N	Mean Rank	Sum of Ranks
Total Duration (mins)	Pakistan	35	33.20	1162.00
	United Kingdom	35	37.80	1323.00
	Total	70		

Test Statistics[a]

	Total Duration (mins)
Mann-Whitney U	532.000
Wilcoxon W	1162.000
Z	-.976
Asymp. Sig. (2-tailed)	.329

a. Grouping Variable: Location

Table 24 - Mann-Whitney U test for duration of conversation between locations

The results of the test highlight that the difference in the time taken to reach the conversation goal between the two evaluation locations was not statistically significant ($p = .329$).

8.15.2 Number of utterances requiring word segmentation between the locations datasets

Table 25 outlines the results of a Mann-Whitney U test carried out in order to determine if there was a statistically significant difference in the amount of user utterances that required word segmentation processing in order to segment the words in the utterances.

Ranks

	Location	N	Mean Rank	Sum of Ranks
Number of utterances with word segmentations	Pakistan	35	41.66	1458.00
	United Kingdom	35	29.34	1027.00
	Total	70		

Test Statistics[a]

	Number of utterances with word segmentations
Mann-Whitney U	397.000
Wilcoxon W	1027.000
Z	-3.304
Asymp. Sig. (2-tailed)	.001

a. Grouping Variable: Location

Table 25 - Mann-Whitney U test for number of utterances with word segmentations between locations

The results show that there is a statistically significant difference in the amount user utterances that required word segmentation between the two countries ($p = .001$). When the mean ranks are compared the results show that Pakistan ranks higher than the United Kingdom, indicating that the conversations from Pakistan contained significantly more instances where the user utterances required processing in order to segment words in to valid words.

8.15.3 Number of unrecognised utterances between the locations datasets

Table 26 displays the results of a Mann-Whitney U test conducted to gauge whether there was a statistically significant difference in the number of unrecognised utterances between the two locations.

Ranks

	Location	N	Mean Rank	Sum of Ranks
Number of unrecognised utterances	Pakistan	35	37.83	1324.00
	United Kingdom	35	33.17	1161.00
	Total	70		

Test Statistics[a]

	Number of unrecognised utterances
Mann-Whitney U	531.000
Wilcoxon W	1161.000
Z	-1.513
Asymp. Sig. (2-tailed)	.130

a. Grouping Variable: Location

Table 26 - Mann-Whitney U test for number of unrecognised utterances between locations

As the p value is greater than .05 (p = .130), it can be concluded that there is no significant difference between the number of unrecognised utterances between the two evaluation locations.

8.15.4 Number of utterances requiring WOW processing between the locations datasets

Table 27 illustrates the results of a Mann-Whitney U test conducted in order to test if there is a statistically significant difference between the numbers of utterances which required processing by the WOW algorithm (i.e. utterances that were word order variations of scripted patterns) between the two locations.

Ranks

	Location	N	Mean Rank	Sum of Ranks
Number of utterances with WOW	Pakistan	35	37.51	1313.00
	United Kingdom	35	33.49	1172.00
	Total	70		

Test Statistics[a]

	Number of utterances with WOW
Mann-Whitney U	542.000
Wilcoxon W	1172.000
Z	-.844
Asymp. Sig. (2-tailed)	.399

a. Grouping Variable: Location

Table 27 - Mann-Whitney U test for number of utterances requiring WOW processing between locations

Since the p- value is higher than 0.05 (p = .399), it can be concluded that there is not a significant difference between the number of utterances which required processing by the WOW algorithm between the two evaluation locations.

8.15.5 Conversation goal achievement between the locations datasets

Table 28 demonstrates the results of a Mann-Whitney U test conducted to gauge whether or not there is a statistically significant difference between the numbers of conversations which met the intended goal of the discussion between the two locations.

Ranks

	Location	N	Mean Rank	Sum of Ranks
Goal achieved	Pakistan	35	35.50	1242.50
	United Kingdom	35	35.50	1242.50
	Total	70		

Test Statistics[a]

	Goal achieved
Mann-Whitney U	612.500
Wilcoxon W	1242.500
Z	.000
Asymp. Sig. (2-tailed)	1.000

a. Grouping Variable: Location

Table 28 - Mann-Whitney U test for conversation goal achievement between locations

The results of the test yielded a p-value of 1.00, thus it can be concluded that there was no significant difference in the number of conversations which reached the intended goal between the two evaluation locations.

This section further explored the data gathered during the evaluation to highlight any differences between the data gathered from the two locations. The following section explores and analyses the questionnaire data that was gathered in order to gauge participants perceptions related to the subjective metrics.

8.16 Analysis of questionnaire data

The questionnaire was split into two distinct parts, the first part consisted of Likert scale questions and the second part related consisted of categorical questions (i.e. Yes/No), both parts aimed to gauge user perceptions with regards to the subjective metrics related to their interaction with UMAIR (see Appendix B for questionnaire). The analysis of the questionnaire data will test and provide evidence towards concluding hypothesis H1-D.

Table 29 summarises the findings of the questionnaire survey from the evaluation of UMAIR with updated architecture and Table 30 outlines a summary of the findings of the questionnaire from the evaluation of the first UMAIR prototype.

Table 33 illustrates the results of the Wilcoxon signed rank test conducted on the matching questions from the questionnaires administered in the old and new UMAIR evaluations.

UMAIR Prototype 2 Likert Scale questions					
	Very Bad	Bad	Neutral	Good	Very Good
User Interface Design	-	-	14.3%	40%	45.7%
System Helpfulness		2.9%	1.4%	58.6%	37.1%
Quality of Information	-	1.4%	2.9%	45.7	50%
Level of System Understanding	1.4%	1.4%	2.9%	37.1%	57.1%
Naturalness of Conversation	-	-	10%	51.4%	38.6%
Level of Satisfaction with Conversation	-	2.9%		37.1%	60%
Time Taken to Reach the Goal	-	-	2.9%	40%	57.1%
Predictive Text Feature	-	2.9%	14.3%	37.1%	45.7%
Categorical questions					
	Yes	No			
Would you use UMAIR again?	98.6%	1.4%			
Would you use UMAIR instead of visiting a NADRA office?	81.4%	18.6%			

Table 29 – Frequency analysis prototype two questionnaire data results

UMAIR Prototype 1 Likert Scale questions					
	Very Bad	Bad	Neutral	Good	Very Good
User Interface Design	4.2%	41.7%	54.2%	-	-
System Helpfulness	-	-	25.0%	75.0%	-
Quality of Information	-	4.2%	12.5%	75.0%	8.3%
Level of System Understanding	4.2%	0.0%	12.5%	75.0%	8.3%
Naturalness of Conversation	4.2%	0.0%	75.0%	20.8%	-
Level of Satisfaction with Conversation	-	-	33.3%	54.2%	12.5%
Time Taken to Reach the Goal	-	-	12.5%	75.0%	12.5%
Categorical questions					
	Yes	No			
Would you use UMAIR again?	96%	4%			
Would you use UMAIR instead of visiting a NADRA office?	42%	58%			

Table 30 – Frequency analysis prototype one questionnaire data results

Statistics

	Design P2	Helpfulness P2	Instructions P2	Understanding P2	Naturalness P2	Satisfaction P2	Time P2	Again P2	Human P2
Mean	4.31	4.30	4.44	4.47	4.29	4.54	4.54	.99	.81

Table 31 –Mean values from evaluation questionnaire two

Statistics

	Design P1	Helpfulness P1	Instructions P1	Understanding P1	Naturalness P1	Satisfaction P1	Time P1	Again P1	Human P1
Mean	2.50	3.75	3.88	3.88	3.17	3.79	4.00	.96	.42

Table 32 –Mean values from evaluation questionnaire one

Test Statistics[a]

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
	Design	Helpfulness	Instructions and information	Understanding	Naturalness	Satisfaction	Time Taken	Use Again	Instead of Human
Mann-Whitney U	65.000	438.000	453.500	416.500	188.000	358.000	441.000	817.000	506.000
Z	-7.012	-4.076	-3.774	-4.085	-6.045	-4.620	-3.900	-.798	-3.695
Asymp. Sig. (2-tailed)	.001	.004	.002	.027	.000	.006	.018	.425	.000

a. Grouping Variable: Evaluation Group

Table 33 – Mann Whitney test between old and new questionnaire data

The first question of the questionnaire was designed to ascertain the participant's perception of the user interface (UI) design. Figure 51 illustrates the results of question 1 from the questionnaire compared to the results of the same question from the end user evaluation of the first UMAIR prototype. The results reveal that the vast majority of the participants (85.7%) rated the UI as 'good' or 'very good'. The figure also illustrates that when compared to the UI of the first UMAIR prototype the updated UMAIR UI was perceived to be better by the participants. These results are further corroborated by the results of the Mann Whitney test carried out on this question in Table 33 that show the difference in perceptions between the two evaluations are statistically significant (p value = .001), with the second iteration of UMAIR's UI having a better perceived response.

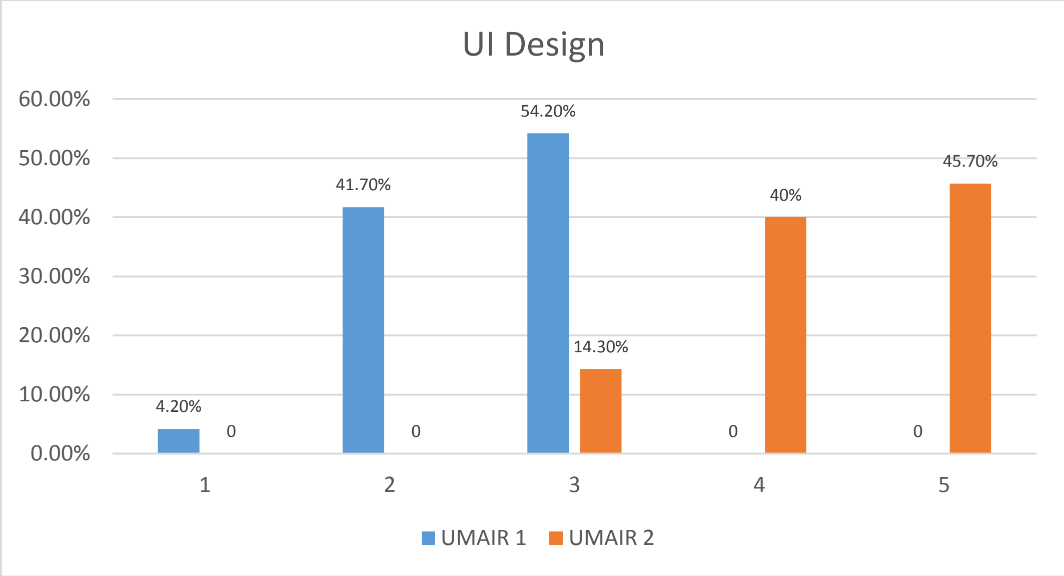


Figure 51 - Bar Chart Question 1 Results

Question 2 of the questionnaire pertained to gauging the participant's perceptions towards the helpfulness of UMAIR. Figure 52 illustrates the results of this question against the results received subsequent to the evaluation of the first UMAIR prototype. The results highlight that there was an increase in number of participants who perceived UMAIR to be 'very good' in helpfulness (37.1%) compared to the first prototype system (4.2%). The results of the Mann Whitney test conducted for this question in Table 33 shows that this increase, and the general perceptions between the two systems is statistically significant (p value = .004).

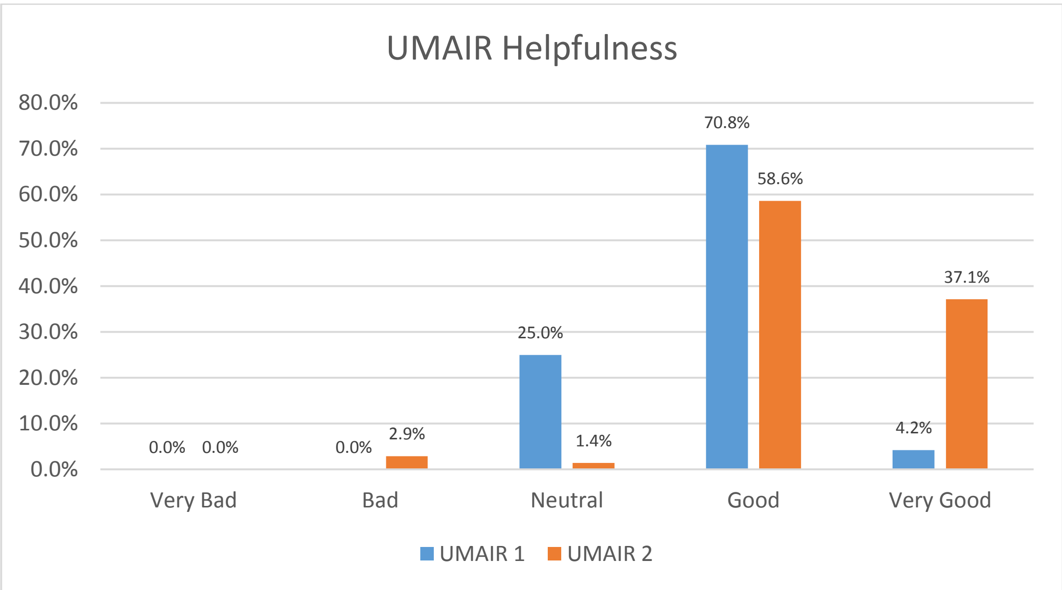


Figure 52 - Bar Chart Question 2 Results

The third question of the questionnaire was designed to gauge the participants perceptions related to the quality of information provided to them by UMAIR during their interaction. Figure 53 outlines is results of this question. The results show that the number of number of participants who expressed that the quality of information provided by UMAIR was ‘very good’ (50%) did increase from the first prototype (8.3%). The results of the Mann Whitney Test conducted on this question in Table 33 reveals that this increase was statistically significant compared to the first evaluation (p value = .002).

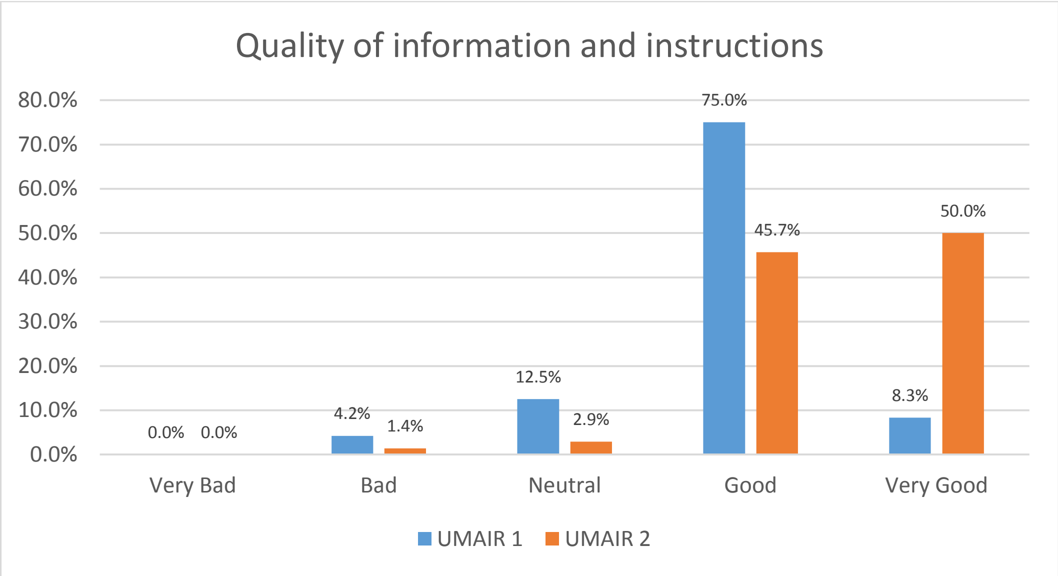


Figure 53 - Bar Chart Question 3 Results

Question 4 of the evaluation questionnaire was designed to ascertain the participants perceptions related to UMAIR’s level of understanding of their utterances and queries. Figure 54 illustrates the results of this question compared to the results data of the same question from the first evaluation. The results show that there is a big increase in the number of participants who rated this question as ‘very good’ (57.2%) compared to the first prototype (8.3%). This results is supported by the results of the Mann Whitney test conducted on this question in Table 33, which indicates that this increase was statistically significant (p value = .027).

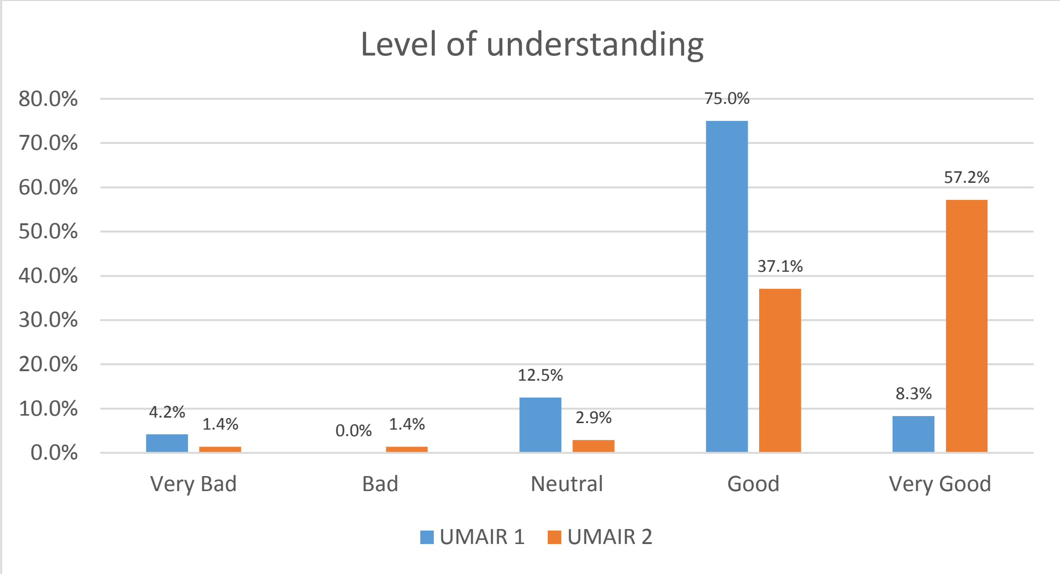


Figure 54 - Bar Chart Question 4 Results

The fifth question of the questionnaire was devised in order to measure the participant's perceptions towards naturalness of their conversation with UMAIR. The results of this question are illustrated in Figure 55 which demonstrates that the user perception towards conversation naturalness increased when compared to the first UMAIR prototype. The findings of the Mann Whitney test in Table 33 also proved that the difference between this result and the first prototype evaluation is a statistically significant (p value = .001) increase. The standout increase in this question is the number of participants who rated the conversation naturalness as 'very good' (38.6%) compared to the number from the first evaluation (0%).

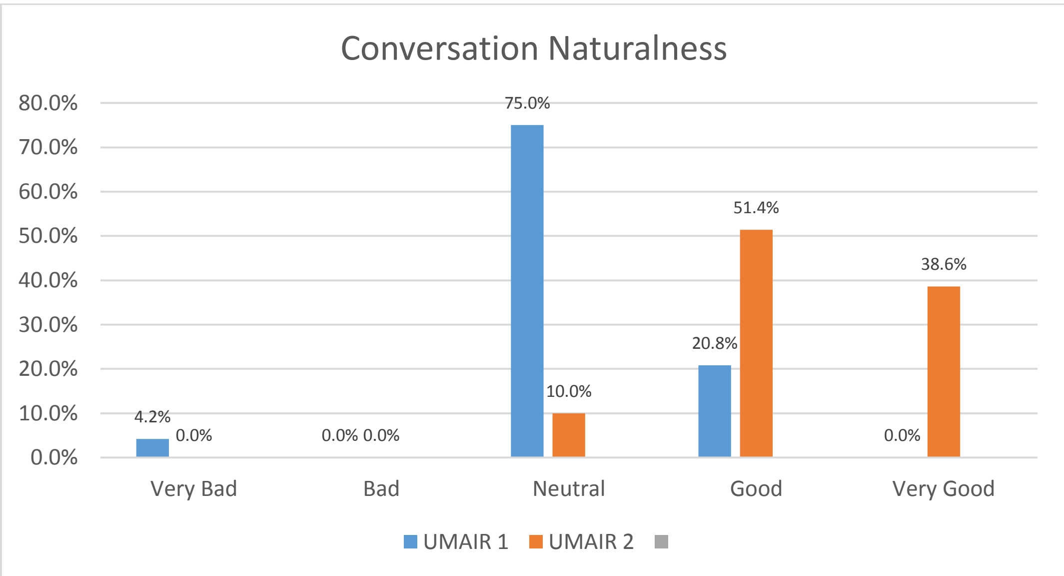


Figure 55 - Bar Chart Question 5 Results

The sixth question of the questionnaire asked the user to rate their level of satisfaction with regards to their interaction with UMAIR. The results of this question are illustrated in Figure 56 which also plots the results of the same question from the first prototype evaluation. It can be seen that the majority of the participants rated their level of satisfaction after having interacted with UMAIR as 'very good' (60%). The results of the Mann Whitney test in Table 33 also show that the difference in the results of this question between the two UMAIR systems was statistically significant (p value = .006).

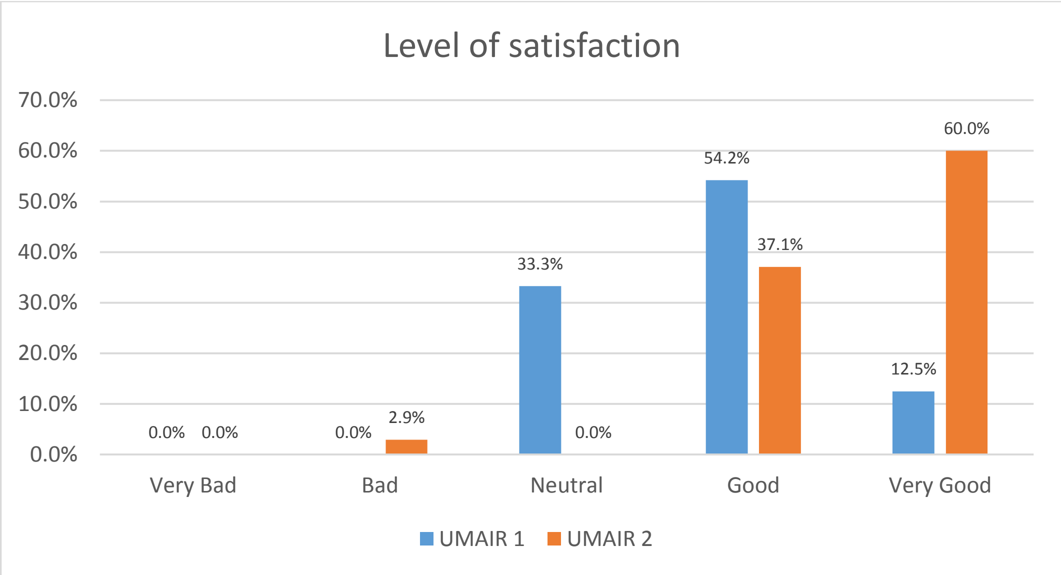


Figure 56 - Bar Chart Question 6 Results

Question seven on the questionnaire was aimed at measuring the participant's perception of the time it took for them to reach the goal or retrieve the information they required from UMAIR. It can be seen from the results outlined in Figure 57 that the majority of the participants (57.1%) rated the time taken to get the information they required as 'very good'. As highlighted in in Table 33, these results proved to be a statistically significant improvement compared to the results of the same question from the first evaluation (p value = .018).

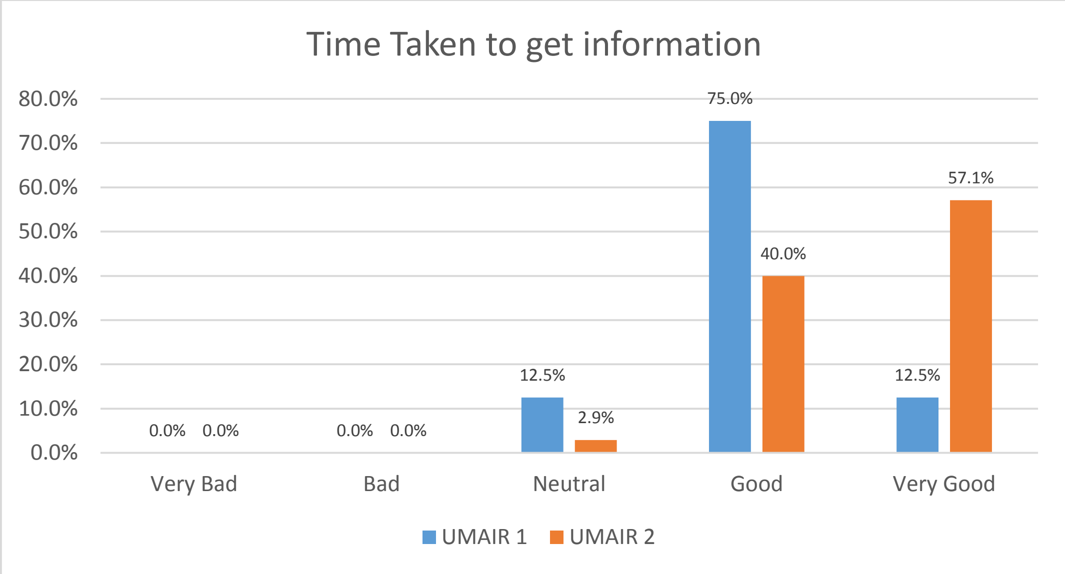


Figure 57 - Bar Chart Question 7 Results

The eighth question of the questionnaire was designed to gather the participant's perceptions with regards to whether or not they would use the system again. The results of this question are illustrated Figure 58 along with the results of this question from the evaluation of the first UMAIR prototype. It can be seen that there was a marginal increase in the participant's perceptions towards this question, with the vast majority of the respondents (98.6%) once again stating that they would use the system again. The results of the Man Whitney test in Table 33 confirmed that the marginal increase was not statistically significant (p value = .425).

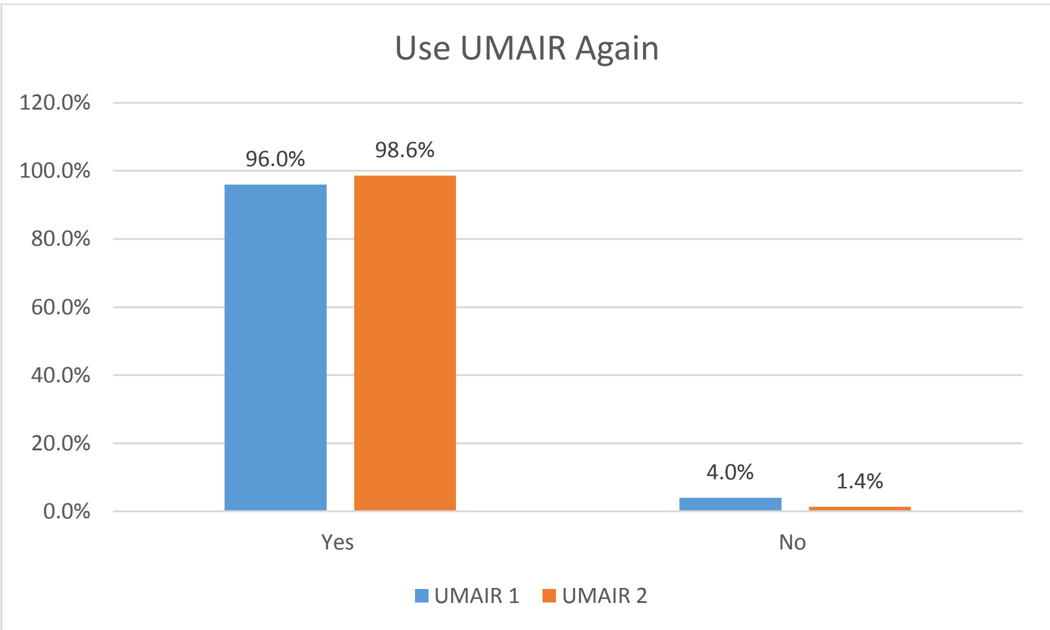


Figure 58 - Bar Chart Question 9 Results

The ninth question of the user evaluation question aimed attest if the participant would prefer to use UMAIR instead of interacting with a human. The results in Figure 59 reveal that there has been a major increase (81.4%) in the number of participants that said they would use the system instead of talking to a human compared to the results of the first UMAIR evaluation (42%). The results of the an Whitney test revealed that the difference in opinion between the participants in the two evaluations was highly statistically significant (p value = .001).

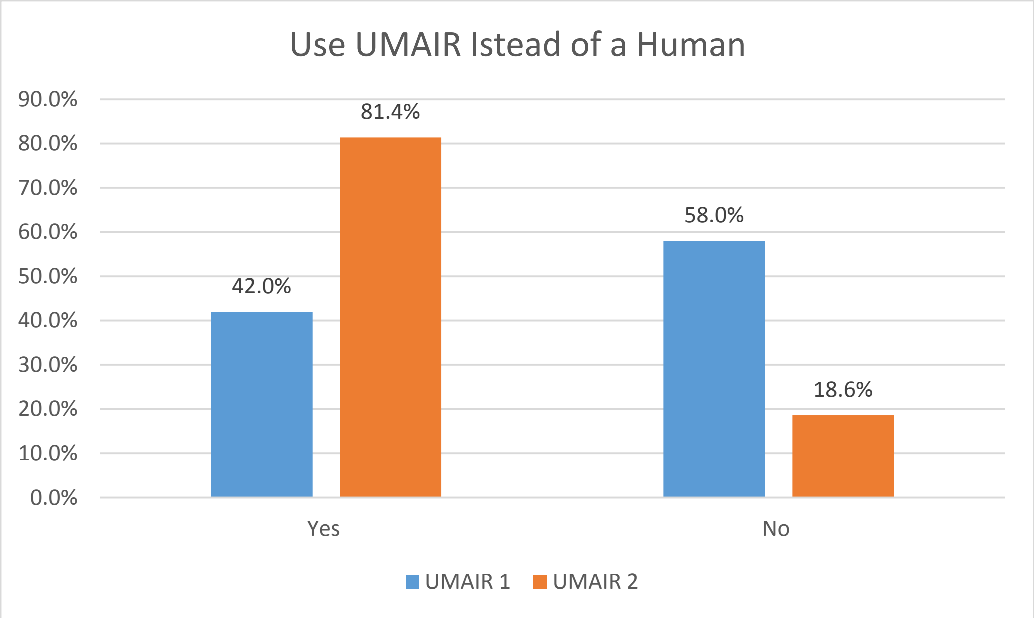


Figure 59 - Bar Chart Question 10 Results

The final question of the user evaluation questionnaire was new, added to the questionnaire specifically to ascertain the success of the predictive text feature that was added to the architecture of UMAIR. The results of this question are illustrated in Figure 60 which shows that the majority of the participant rated the predictive text feature as either ‘very good’ (45.7%) or ‘good’ (37.1%).

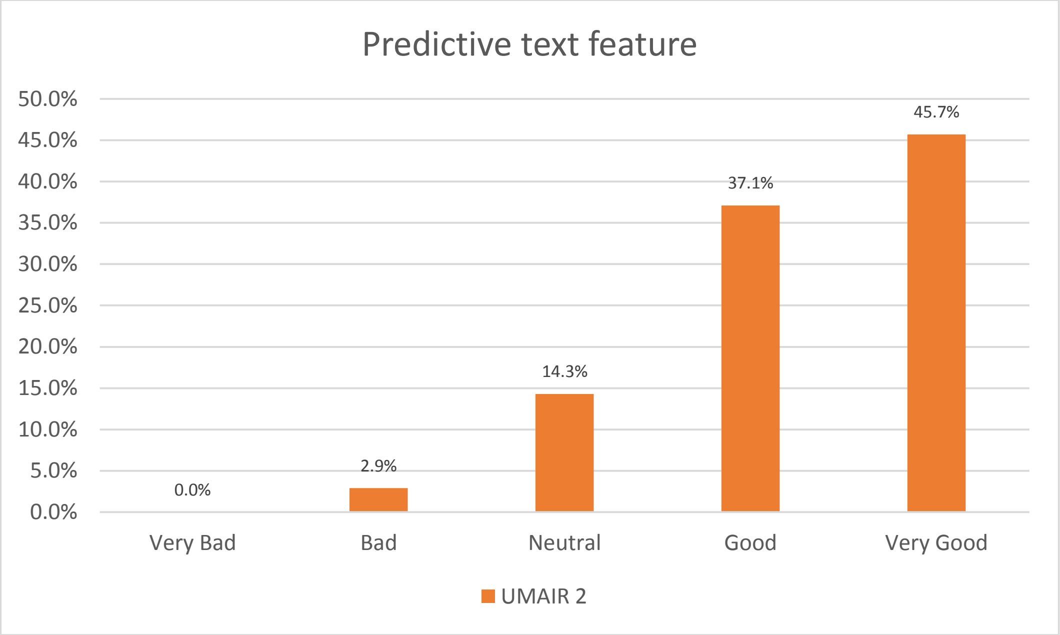


Figure 60 - Bar Chart Question 8 Results

The final questionnaire was an open question to the participants which asked them if they had any additional comments with regards to their experience with UMAIR. Many of the participants responded to this question with praise and positive comment for UMAIR, a few such comment cited below:

"The little animated man was very nice, I like the way he pointed at the part of the screen to help me"

"Quick and straight to the point, very cool"

"I enjoyed this much more than my last visit to the Manchester NADRA office, very straight forward"

"This is a good thing for Pakistanis that live far away from the NADRA offices, especially them who live in remote villages" [Sic]

"I enjoyed talking to UMAIR, he even responded to my silly questions like the weather in a humours way! This is good for people who cannot communicate in English like my parents"

Some of the participants responded with some comments that related to how they thought UMAIR could be improved, they made the following points:

"I think adding voice based interaction will make this system much better"

"When typing the word suggestion should display more suggestions not just one" [Sic]

"More interaction from the man would be good, I like how he reacted to my questions"

"I think UMAIR should cover all NADRA related questions not just ID card and Passport. Also I think that UMAIR should be expanded to cover all government department related questions like tax and housing etc."

8.17 Results Conclusion

The results demonstrate that the enhancements made to UMAIR's architecture in prototype 2 have made a statistically significant impact on the effectiveness of UMAIR's engine when compared to prototype 1. The results highlighted, in certain key metrics that related to effectiveness and accuracy (i.e. number of unrecognised and goal achievement) prototype 2 performed better than prototype 1 from an objective perspective. These results suggest that the enhancements made to the components and newly added components were relevant to the improvement of the effectiveness, accuracy and robustness of UMAIR.

Furthermore, the results show that the new components added to UMAIR's architecture have resulted in fewer unrecognised utterances, indicating that the improvements made the WOW algorithm, the addition of the word segmentation/validation and the predictive text features have had the intended impact on the engine. These components come together to make UMAIR's engine more robust and effective, which when compared to the prototype one has led to a significant increase in the conversation goal achievement.

The key finding of the results of the statistical analysis also revealed that the second UMAIR prototype had significantly fewer unrecognised utterances when compared to the first prototype.

Furthermore, the results also demonstrated that the second prototype showed an improvement in the metrics related to the subjective perspective of the evaluation. The results highlighted that the participants perceived the second prototype to be better, notably the weaknesses highlighted during the evaluation of the first prototype for example, naturalness of conversation, system understanding and user interface design were perceived to be better in the second prototype. Moreover, all metrics that were evaluated through the questionnaire demonstrated an improvement in participant perception.

Additional interesting findings that stood out from the statistical analysis of the evaluation data included the differences found in user interaction between the two evaluation location datasets. The comparison of the data from the two location datasets illustrated that the utterances from the participants in Pakistan contained significantly more instances where the utterance contained words that required processing to split words in to valid words.

The most notable point arising from the analysis of the questionnaire, is in the comparison of the participant's perceptions from the first prototype the participant's perceptions towards the second UMAIR prototype revealed that the second prototype was perceived to be better in all aspects (e.g. helpfulness, quality, UI design etc.). The second prototype was received more enthusiastically by the participants notably where the first prototype was lacking, specifically in the areas of UI design, helpfulness, quality of information and instructions, level of understanding, naturalness of conversation and user satisfaction.

All areas showed an improvement in participant's perceptions which is evident from the mean values of the questionnaire outlined in Table 31 and Table 32, indicating that the improvements and enhancements made to UMAIR collectively had a positive impact on UMAIR's engine and furthermore the user experience. From the comments received in the open question to the participants it can be seen that the respondents enjoyed their interaction with UMAIR and more importantly the participants perceived UMAIR as a useful tool for getting information about the ID card and passport application process.

The implications of these findings and results on the research hypothesis are discussed in the following chapter.

Chapter 9 - Discussion

The second prototype was developed through further research and development in order to address the weaknesses highlighted through the evaluation of the first prototype. The main weaknesses highlighted during the first prototype evaluation were the robustness and accuracy of the engine. The engine had weaknesses due the unique morphological features of the Urdu language such as spelling variations and inconsistent word segmentation. These language unique challenges had a detrimental impact on the accuracy and robustness of UMAIRs engine. Furthermore the end user evaluation of the first prototype revealed some further weaknesses through the evaluation questionnaire. The questionnaire unearthed some user perceived weaknesses. The main weaknesses perceived by the users was the naturalness of the conversation, user interface design and the level of understanding of the system. These weaknesses were addressed through further research and development, which lead to the addition and strengthening of components in UMAIR’s engine in order to diminish these weaknesses, which then became the second prototype of UMAIR.

The primary aim behind the second evaluation was to gauge whether or not the new components added to UMAIRs architecture had any impact on the effectiveness, accuracy and general performance of UMAIR’s ability as a conversational agent. The results of the second evaluation outlined in the previous chapter, reveal that the second UMAIR prototype performed significantly better compared with the first prototype in relation to the objective and subjective metrics measured between the two UMAIR prototypes. The second prototype was better in terms of objective task completion and in relation to end user perceptions.

The hypothesis that were tested through the second evaluation and their results are as follows:

H1- the enhancement made to UMAIRs architecture improve the overall effectiveness and robustness of UMAIRs engine.

> **H1-A.** The improvements made to the WOW algorithm have an impact on the accuracy and effectiveness of UMAIRs engine and reduces the percentage of unrecognised utterances.

H1-B. The addition of the word segmentation feature made an impact in improving UMAIR's engine in terms of reducing the rate of unrecognised utterances.

H1-C. The addition of the predictive text feature made an impact in improving UMAIR's engine by reducing the rate of unrecognised utterances.

H1-D. The improvements made to UMAIR's result in better perceptions from the users in relation to the subjective aspects.

The main hypothesis (H1) is accepted or rejected based on the results of subsidiary hypothesis. When looking at the improvements made to the WOW algorithm which were adopted to recognise common spelling variations. The lexical similarity algorithm (i.e. Levenshtein) was redeveloped specifically for Urdu, it now allows the common variations of certain words to be recognised and responded to correctly through compensating for phonetically similar characters (see chapter 7 section 7.2). Furthermore the WOW algorithm was further strengthened by including the number of matching words in the pattern and utterance in the similarity calculation, in order to make it more accurate and reduce occurrences where the incorrect rules fired. These improvements have reduced the number of unrecognised utterances which is evident in the results.

The evaluation results taken from the log file reveal that incorrect rules firing was reduced to 3.17% (Table 19) in the second prototype compared to 12% in the first prototype. These results are substantiated by the results of the statistical analysis of the log file data, which compared the number of utterance processed by the WOW algorithm (Table 22). The results of the test revealed that the number of utterances that required processing by the WOW algorithm between the two systems were not found to be statistically significantly different. That means the number of user utterances that required processing by the WOW algorithm was not different. However, since the results also show that the number of incorrect rules firing was significantly reduced in the second prototype, it suggests that in the enhancements made to the WOW algorithm had a positive effect on the accuracy and robustness of the engine in the second prototype. Therefore, based on these results there is evidence to **support H1-A**.

The Word segmentation algorithm was developed through the findings of the evaluation of the first UMAIR prototype. The results of the first evaluation highlighted the fact that some of the participants involved in the evaluation of the first UMAIR prototype were utilising one of the language unique features of Urdu, which was the feature which allows the omitting of space in written text. As discussed in chapter 3 section 3.8 the use of a white space as a separator is not a consistent method to determine word boundaries in Urdu. This language unique feature had detrimental effects on the robustness and accuracy of UMAIR's engine. In light of this, a word segmentation algorithm was researched and implemented in to the second prototype of UMAIR's engine to pre-process the user utterances to ensure the words in the utterance were correctly segmented. The results of the second evaluation reveal that the word segmentation algorithm played a major role in reducing the number of unrecognised utterances. A total of 5.96 % (32 utterances) of utterances in all conversations contained instances where words required segmentation by the algorithm. Without the word segmentation algorithm these utterances would have resulted in the engine failing to recognise the utterances. These results are corroborated by statistical analysis of the log files, which compared the number of unrecognised utterances between the two UMAIR prototypes (Table 20). The results of the test revealed a statistically significant difference between the numbers of unrecognised utterances, with the second prototype having a lower mean rank of unrecognised utterances. Hence based on these results there is enough evidence to suggest that **H1-B can be accepted**.

The predictive text input feature was introduced in the system in order to reduce spelling mistakes made by users when entering text/utterances into the system. The result of the first prototype evaluation revealed that a large proportion of the unrecognised utterances were due to spelling mistakes made by the user. The results of the questionnaire revealed that the majority of the participants (82.8%) expressed that they thought the predictive text feature was a useful feature. The results of the second prototype reveal that the word segmentation as well as the predictive text typing feature worked towards increasing the accuracy and robustness. Collectively the addition of these features point towards a significant improvement in the effectiveness of the UMAIR's engine, thus the evidence supports **accepting H1-C**.

In H1 A, B and C the evaluation measures support the improvement, and the log file analysis demonstrates the contribution of the factor. In the case of H1 A. B and C the results show that each of the components added to address the shortcomings of the first prototype are all factors that are contributing positively towards increasing the accuracy and robustness of UMAIR's engine.

Additional findings of interest derived from the evaluation of the second prototype were highlighted through the comparison of the data gathered from the two different countries of evaluation. The comparison of the data gathered form the two countries revealed some significant differences between the ways the participants from the two countries interacted with the system. Most prominent of the differences observed was the utterances of the participants in Pakistan contained statistically significantly more instances where the utterance contained words that required processing by the word segmentation algorithm to split the words in to valid words. A reason for this could be that the participants who tested the system in the UK also use English on a daily basis with its space separation scheme, therefore when these participants entered information they were more consistent in use spaces for word segmentation as compared to the participants from Pakistan who use Urdu as their main language for communication and consequently are not as consistent with their use of white space to separate words as in Urdu it is optional depending on the context. This result highlights the significance of the word segmentation algorithm as without it all the utterances that required word segmentation would not have been recognised by the engine, which would have had a detrimental effect on the accuracy and robustness of UMAIR's engine.

Inevitably as with any research and evaluation effort the evaluation of the second prototype did highlight some areas of UMAIR that can be improve through further research. Firstly the knowledge base was found to have some gaps in its domain knowledge, and general knowledge which led to some unrecognised utterances. However, these gaps are easily addressed as the missing information can be added to the knowledge base through scripting more rules that occur most frequently which will bolster the agent's knowledge. This type of weakness is expected as it is unreasonable to expect the scripter of the knowledge base to anticipate every question that could be asked by the user. Furthermore, in relation to the general knowledge there is a limitation to how much knowledge can be added to make the CA seem more intelligent. The creation and maintenance of a knowledge base is an iterative and ongoing task, as such these shortcoming are expected. The majority of unrecognised utterances during the evaluation were caused by out of context discussion from the users which lead to UMAIR responding to the user stating ‘sorry I didn’t understand you, could you say that in another way’. Other instances of unrecognised utterances stemmed from the users using slang and other language that was not scripted in the knowledge base. However, a positive that can be drawn from this result is that even in the conversations that contained unrecognised utterances the majority of them did eventually lead to the goal of the conversation, which indicated the increased robustness of UMAIR’s engine and its ability to handle unrecognised user utterances.

UMAIR is able to converse with the user in order to lead the conversation towards a goal. UMAIR leads the conversation and ensures the user stays on the path in order to reach the goal. UMAIR is also intelligent enough to be able to handle out of context discussion, when the user changes the topic of the discussion to something other than the current context/topic UMAIR responds by answering the question/utterance that is out of the current context then steers the conversation back towards the goal. An example of this is illustrated in Table 34.

Topic: Lost Passport				
		English translation	**Urdu Utterance**	**Engine Attributes**
1	UMAIR	Hello my name is Umair, welcome to the NADRA online helpdesk. I can answer queries regarding to ID card and passport applications. How can I help you today?	سلام میرا نام عمیر ہے. خیر مقدم. میں شناختی کارڈ اور پاسپورٹ کی درخواست میں آپ کی مدد کر سکتے ہیں. میں آپ کی کیسے مدد کرسکتے ہیں؟	Searching for context
2	User	Hello umair brother, how are you.	وعلیکم السام عمیر بھائی ، تم کیسے ہو؟	Greeting
3	UMAIR	I’m fine thank you, how can I help you?	میں ٹھیک ہوں شکریہ، میں تمہیں کس طرح مدد کرسکتے ہیں؟	Searching for context
4	User	Brother I have lost my passport, and I am traveling in a few weeks to China.	بھائی میں نے اپنا پاسپورٹ کھو دیا ہے، اور میں نے چین کے لئے چند ہفتوں میں سفر کر رہا ہوں.	Context found: **lost passport**
5	UMAIR	OK, in order for you to get a new passport you will need some proof of identification such as a birth certificate or an ID card.	کوئی مسئلہ نہیں، آپ کو شناخت کے ثبوت ہے؟ شناختی کارڈ یا پیدائش کا سرٹیفکیٹ؟	Initialisation rule for lost passport path
6	User	Yes I have a ID card	جی ہاں اپنا شناختی کارڈ ہے	Path followed

7	UMIAR	Excellent, In that case you need to fill out a PA form, have you filled in one of these?	عمدہ، آپ کو ایک PA فارم پر کرنے کی ضرورت ہے. آپ نے فارم بھر دیا ہے؟	-
8	User	But brother, how long does a passport application take?	پر بھیا، کتنی دیر تک ایک پاسپورٹ کی درخواست لگتا ہے؟	Path not followed context found: **FAQ**
9	UMAIR	The normal time it takes for NADRA to process a passport application is 2 – 3 weeks, but you didn't tell me, have you filled out a PA form?	یہ عام طور پر 2 یا 3 ہفتے لگتے ہیں، لیکن ، آپ نے PA فارم بھر ہیں کیا؟	Answer FAQ and direct conversation back to path
10	User	Right then I should have enough time, no I didn't fill out the PA form. I'll do that today.	یہ بات اچھی ہے، میرا پاس وقت۔ نہیں میں آج PA فارم کروں گا	Path followed
11	UMAIR	Right, you must fill out a PA form, attach the correct fee for a passport and submit it to your local NADRA office. Is there anything else I can help you with?	ٹھیک ہے، تم ایک PA فارم بھرنا چاہیے، صحیح فیس کے ساتھ، آپ کو مقامی نادرا دفتر پر بھیج دیں۔ اور کچھ؟	Conversation reached goal. Ask for further assistance
12	User	Is it possible to fast track a passport application?	میں اسے جلدی کروا سکتا ہے؟	Context found: FAQ
13	UMAIR	Yes, you can find more information about application duration charges on the following web site: http://www.nadra.gov.pk/index.php/products/cards/cnic Is there anything else I can help you with?	جی ہاں، آپ کو مزید معلومات یہاں مل سکتے ہیں: http://www.nadra.gov.pk/index.php/products/cards/cnic کچھ اور تھا؟	Answer FAQ and ask for further assistance
14	User	No that's all thanks.	نہیں، شکریہ۔	Context found: conversation terminator

Table 34 - Translated example of discussion from log file

The example illustrates how UMAIR responds to situations where the user leaves the predefined path and ask a question out of the current context/flow of discussion, the out of context question (utterance 8) is answered and then the user is asked the previous question again.

A questionnaire was utilised to capture data with the intention of shedding light on UMAIR from the participants subjective perspective. The results of the questionnaire demonstrated a significant improvement in the participant's perceptions towards the second UMAIR prototype. The results from the end user questionnaire administered after the evaluation of the first UMAIR prototype revealed that participants perceived the conversation with UMAIR to be low in naturalness. The participants stated that they thought that the first prototype of UMAIR was repetitive and robotic in its interaction. In order to address this issue the second prototype included three key developments. Firstly more general knowledge not related to the domain was added to the knowledge base so UMAIR could respond to more general utterances. Secondly, several responses to each rule was added to the knowledge base so the same response to a fired rule was not always delivered back to the user. This was done in order to provide some variation in the responses delivered to the user, making the conversation less repetitive when the same rule was fired multiple times during a conversation, which helped make the conversation more natural. Furthermore, short term memory was added to the architecture in order for UMAIR to recall previously fired rules to simulate a short term memory, and respond to repetition more naturally. The results of the questionnaire from the evaluation of the second prototype revealed that the users perceived the second UMAIR prototype to be better in relation to conversation naturalness with 90% of the participants expressing that the conversation level of naturalness was either good or very good which is a major improvement from the first prototype where only 20% fell in into these two categories. This result indicates that the developments made a positive impact on the naturalness of the conversation between UMAIR and the participants.

In addition to the naturalness of conversation, the design UI also received negative feedback from the participants of the end user evaluation of the first prototype who stated that it was bland and uninteresting. In order to address this a different approach to CA UI design was adopted that included a small embodied character. The results of the evaluation from the second prototype reveal that the majority (87.5%) of the participants expressed that the UI design was either good or very good compared to the evaluation of the first prototype where none (0%) of participants expressed that they thought the UI design was either good or very good. From these results it can be seen that the inclusion of the embodied character has had a positive impact on the user perceptions related to the UI. These findings coincide with the findings of Cassell et al. (2001), who state that the embodiment of a CA that illustrated nonverbal behaviour can enrich the end user experience and improve end user perceptions (Bickmore and Cassell, 2005). The number of users rating the system understanding and level of understanding as 'very good' significantly increased this finding is a result of the improvements made to the knowledge base, which resulted in fewer occurrences where UMAIR failed to understand the user utterances. Therefore, based on these and the other results gathered from the questionnaire it can be concluded that there is enough evidence to **accept H1-D.**

To conclude, in relation to improving the effectiveness of UMAIR with regards to objective task completion the aim was to reduce the number of unrecognised utterances and increase the rate of goal achievement of the conversations with UMAIR. The results revealed that compared to the first prototype the second prototype was better in both aspects (i.e. less unrecognised utterance and more conversations leading to goal achievement). The second prototype had 3.17% of unrecognised utterances which proved to be statistically significantly less than the 12% of the first prototype. Moreover,5 the second prototype had 97% of conversations leading to the goal of the discussion which was a significant improvement to the first prototype which had 83.3% of conversations leading to the goal of the discussion.

Based on these results from the objective and subjective metrics measured through the evaluation of the second UMAIR prototype it can be concluded that there is enough evidence to support **H1**. The newly researched and developed components and enhancements point toward an overall improvement in the objective metrics gauged in UMAIR's engine and an increase end user perceptions in relation to the subjective metrics.

Chapter 10 - Thesis Conclusion

This thesis has presented research into the development of an Urdu CA. The research endeavour entailed thorough investigation in to several key areas of CA development namely, CA's, Language Processing techniques (i.e. natural language processing, sentence similarities measures and pattern matching), thorough research in to the Urdu language, and CA evaluation methodologies with the intention of developing an effective, functional Urdu CA. The Urdu language is inherently different in grammar, structure and syntax when compared to English, therefore existing CA engines were not suitable to process Urdu text. Due to the nature of the Urdu language the research into CA development techniques revealed that the pattern matching (PM) approach was the most appropriate approach to adopt to develop an Urdu CA. This led to the development of UMAIR an Urdu CA. UMAIR's engine is a rule base engine that is comprised of several novel components in order to process the Urdu language. The components include a hybrid engine which is based on the two main CA development strategies, A PM engine and a lexical string similarity (WOW) component that calculates the matching strength of a pattern to the user utterance without taking into consideration the semantics of the utterance. The two parts of the engine work together in order to alleviate some of the language unique challenges of the Urdu language. Due to the challenges the Urdu language posed in its implementation within a CA, the research also led to the development of additional novel components which were implemented in UMAIR's architecture in order for the language to be able to be processed accurately. One such component was the word segmentation algorithm, which was researched and developed in order to mitigate a language unique issue of in consistent word segmentation posed by the Urdu language. Furthermore, a novel Urdu scripting language was developed that encompasses many new features like the ability to work with knowledge trees, which works together with the new engine and architecture to deliver a coherent and intelligent conversation to the user. PM conversation agents are a popular method for developing CA's, however CA's based on the PM principle face criticism and disadvantages in the number of patterns that have to be scripted in order to create a coherent and robust knowledge base. The research discovered that this disadvantage is further exacerbated when implementing an Urdu conversation agent due to the nature of Urdu grammar and its free word order. However, the research of UMAIR led to the development of the WOW algorithm presented in this thesis. The WOW algorithm was researched and developed in order to reduce the effort required in scripting the knowledge base/domain. The algorithm finds word order variations of scripted patterns during run time and matches them to the user utterance, therefore alleviating the need to script all possible word order variations of that pattern in the knowledge base.

Moreover, the evaluation of the first UMAIR prototype brought to light certain unforeseen issues that were unique to the Urdu language. The most prominent issue was word segmentation which had to be addressed in order to produce an effective Urdu conversational agent with a high degree of accuracy and robustness.

In light of the revelations from the first evaluation further research was conducted in order to address the shortcomings brought to light, this researched formed the second prototype. Through the research several new components were developed and enhancements/amendments to existing components in UMAIR's architecture ensued all in the effort to increase UMAIR's effectiveness, accuracy and robustness. The results of the end user evaluation for the first UMAIR prototype revealed some weaknesses/negative perceptions from the participants. The participants expressed that they perceived the naturalness of their conversation with UMAIR to be low, meaning that they felt it was not as natural as talking to a human. Another point revealed from the questionnaire was the participant's perception of UMAIR's UI. The participants expressed that they disliked the UI implemented in the first prototype of UMAIR.

From the body of acquired results through the second evaluation it can be deduced that the amendments made to the second prototype of UMAIR's engine in order to improve the effectiveness, accuracy and robustness of the engine are successful. The results from the evaluation of the second prototype illustrate statistically significant improvements in terms of the quantitative objective metrics measured. The second prototype had a significantly better conversation success rate, meaning more of the conversations led to goal achievement, also the accuracy of the engine was improved significantly due to the amendments made to the WOW similarity algorithm, the addition of the word segmentation algorithm.

Furthermore, results of the end user questionnaire from the second prototype revealed that the participant perceived the second prototype to be better in all metrics measured.

The end user evaluation questionnaire for the second prototype revealed that all of the subjective metrics measured through the questionnaire saw an improvement in relation to how they were perceived by participants. Indicating that the changes and improvements made to address the shortcoming found in the first evaluation had the desired effect on the subjective metrics measured.

UMAIR is able to converse with the user in order to lead the conversation towards a pre-determined goal. UMAIR leads the conversation and ensures the user stays on the path in order to reach the goal. UMAIR is also intelligent enough to be able to handle out of context discussion, when the user changes the topic of the discussion to something other than the current context/topic UMAIR responds by answering the question/utterance that is out of the current context then steers the conversation back towards the goal. This is only made possible through the novel researched and developed components/algorithms that are specifically designed to address the language unique challenges posed by Urdu.

The research aimed to answer the question, can the Urdu language be implemented in a CA to produce an effective, functional CA? The term effectiveness was researched for its relation to software development and broken down in to two distinct perspectives which were the objective and subjective sides of software design and evaluation. Given the challenges that were faced and the results observed from the evaluations, the weight of the evidence supports the conclusion that the CA developed (i.e. UMAIR) is effective as a CA. The researcher was limited in development choices as the state of language processing research in the Urdu language (or indeed any non-western language) is still in its early stages and not as established as research in to western languages such as English.

Nevertheless, the research led to some novel contributions which filled some distinct gaps in the field of CA development such as the WOW and word segmentation algorithms, a new framework for CA development and a new generalised framework for the evaluation of CA's. The new algorithms have mitigated some of the main challenges posed by the Urdu language. The WOW algorithm can theoretically be applied to any language with free word order as it is based on PM principles, consequently languages with free word order such as Arabic and Hindi can utilise it to reduce the scripting effort when developing and implementing CA's in these languages. The word segmentation is a proof of concept that demonstrates how a complex language like Urdu necessitates the user utterance to be pre-processed in order for the PM engine to be able to work more efficiently and accurately. This sort of complex pre-processing is not required in a language like English as the syntax and grammar rules in English are more ridged and strict. Whereas a language like Urdu has a less strict grammar rules, which has a major impact on the development of language processing applications as there are more challenges that are unique to that language to consider.

Moreover, in a language like Urdu where the word segmentation and spelling of words can be inconsistent, the developers of a language processing applications must handle such challenges. Therefore, if these points are overlooked then they can prove to have detrimental effects on the accuracy and efficiency of the language processing engine. Through the end user evaluations it was found that the user exploited the word segmentation rules of Urdu wherever possible. Therefore the developers of CA cannot tangibly expect user to leave consistent word segmentations, which is something that is taken for granted in English and other western language processing applications.

In the early stages of this research many challenges were identified due to the inherent differences of the Urdu language and the current state of the Urdu language research and lack of resources. As a consequence of this research a functional Urdu CA (UMAIR) has been developed which mitigates many of the identified challenges relating to the language and the lack of computational resources. This answers the research question that it is indeed possible to produce an effective and functional CA in the Urdu language. Since the foundational work of Urdu language CA’s has been addressed through this research endeavour, further work in the field of Urdu CA development can build atop of this work which is discussed in section 10.2.

10.1 Research Contributions

This research endeavour has produced some significant academic and practical contributions in the field of Conversational Agent development and language processing. The main objective of researching and developing a functional/effective Conversational Agent in the Urdu language lead to the development of UMAIR. The research and development of UMAIR inevitably lead to the discovery of language specific issues that had to be overcome in order to develop a functional and effective Urdu language CA. The language challenges addressed, algorithms, development and evaluation/testing methodologies derived from this research form the basis of new knowledge contributions that can be utilised as a starting point by future researchers and practitioners in the field in order to research and develop and test and evaluate CA's in other languages.

The concepts, frameworks, methodologies and algorithms presented are language and domain independent. Thus allowing future researchers to utilise them as per their requirements. The prominent contributions derived from this research are as follows:

10.1.1 Urdu CA engine

The research has led to the development of a novel prototype CA engine based on pattern matching principles which incorporates new algorithms for processing user utterances and calculating string similarity in order to converse with the user to reach the goal of the conversation. The Urdu engine handles the language unique features of the Urdu language (as outlined in chapter 4 section 4.2.1) i.e. free word order, ambiguity through diacritics, inconsistent word segmentation. The language specific features found in Urdu do not have to be addressed in existing English CA engines as the grammar and morphological structure is completely different to Eastern languages such as Urdu. Thus, this research makes a contribution in terms of a framework for developing a CA engines in language other than English, and a methodology which can be utilised as a roadmap by future researchers to develop language specific CA engines in other languages.

10.1.2 Urdu scripting language

A new Urdu scripting language with new parameters and measures has been developed in order to script the domain. The scripting language works together with the Urdu engine in order to process the Urdu language and mitigate some of the language unique issues. The scripting language proposed in this research contains new features that deal with the unique features of the language such as allowing word order variations of certain patterns. Other variables in the scripting language allow scripts to store links to accompanying media and documents to support and enrich the conversation and variables that work with the similarity algorithm in order to calculate the pattern strength. Moreover, the scripting language also incorporates variables which allow it to work with the decision trees, these variables are used by components in the engine to allow UMAIR to control the flow the conversation.

10.1.3 WOW algorithm

The WOW algorithm reduces the need for excessive scripting, which is a challenge that comes with the Urdu language and is a unique issue specific to languages with free word order and their implementation into CAs. The algorithm complements the PM method by allowing minimal scripting in order to extract maximum PM information from each scripted pattern. The algorithm improves script maintenance and rule misfire as less patterns have to be scripted, it also improves the overall robustness of the CA engine as it incorporates a new similarity calculation to calculate the similarity between patterns and utterances. The algorithm can be used by future researchers who are developing CA's in other languages with free word order as this language phenomenon is not unique to Urdu it is also found in Arabic and Thai.

10.1.4 Urdu word segmentation algorithm

A proof of concept word segmentation algorithm has been researched, developed and implemented into the engine of UMAIR which was designed to tackle another language specific issue of the Urdu and morphologically similar languages. The word segmentation algorithm allows the user to enter utterances without spaces which is a feature of language that is present in Urdu, Arabic and Farsi as discussed in chapter 3 section 3.8. The word segmentation algorithm then processes the utterances in order to split words in the utterances into valid words, which can then be processed by the similarity algorithm. The word segmentation algorithm allows the maximum information to be extracted from the user utterance by ensuring the words are correctly segmented so they can be processed by the engine. The results of the evaluation highlighted the necessity for the word segmentation algorithm, without which the accuracy and effectiveness of an Urdu CA is reduced.

10.1.5 Methodology for CA development in resource poor languages

A generic CA development methodology has been devised that can be utilised by future researchers and practitioners in order to develop CAs in other languages that have poor linguistic computational resources. This methodology can followed by future practitioners in order to develop new CA engines in languages which differ in morphology and grammatical structure to English. The methodology used in this research can provide a foundational framework which can be utilised and adapted to suit the unique challenges that may arise in other languages. As demonstrated in the UMAIR implementation.

10.1.6 Framework for CA evaluation

A new CA evaluation framework has been researched and tested which addresses the gap in current research related to the development and subsequent evaluation of natural language systems in general. The framework comprises of CA evaluation from an objective as well as subjective perspective in order to give an overall performance related CA evaluation. The proposed methodology focuses on evaluating metrics related to the CA's ability to achieve the goal of its development by employing software evaluation methodologies such as the Goal Question Metric (GQM). This approach allows the CA to be tested on an individual basis, meaning the metrics that are tested from system to system are derived based on the context of the systems implementation, thus allowing the evaluation metrics to be different depending on the development goals of the system being tested. This methodology can be utilised by future research and practitioners to evaluate developed CAs, as the methodology is adaptable to suit individual CA development goals.

10.2Future research

The research presented in this thesis has outlined a novel approach to conversation agent design in a language which is resource poor and completely different to English in its grammatical and morphological structure. However, the implemented framework is not a definitive answer to all the challenges posed by the Urdu language, there are areas which can certainly be improved through further research and development. Possible avenues of further research that could be undertaken to further strengthen and improve the architecture of UMAIR are detailed below.

Semantic similarity

As and when a suitable Urdu WordNet is available the WOW similarity algorithm can be strengthened considerably with the addition of semantic rather than lexical similarity. The addition of semantic similarity will allow the engine to recognise and identify a paraphrased version of a scripted pattern which will further reduce scripting down to a few prototype sentences. Furthermore, as the WOW algorithm is able to recognise word order variations, the addition of semantic similarity will allow the engine to recognise word order variations and semantic variations of user utterances. Therefore, making the task of scripting an Urdu CA even less exhausting.

However this will make the engine more susceptible to making incorrect matches and increase rule conflict so research will be required on how semantic similarity can be combined with the WOW similarity algorithm to make the matching more accurate and robust.

Voice recognition

The addition of voice recognition will make UMAIR or any conversational agent more accessible to a wider audience as well as those people who cannot use a computer (i.e. people who are not computer literate or people with disabilities). Furthermore, the implementation of an Urdu voice recognition will reduce spelling and other user related errors that occur from the users typing text manually to interact with the CA. This may also reduce the burden on other components by reducing computational complexity therefore contributing to scalability on a large saleability web deployment.

Dynamic knowledge base creation

Another interesting direction that future research could take is researching and developing a methodology/technique for the dynamic creation of a CA knowledge base from recorded audio taken from call centres and other customer service areas. The idea is to take the recorded audio, identify/tag the user utterances and the customer service agent response, extract the audio and convert it into text which can then be stored in a structured knowledge base that can be utilised by the engine to find matches to user utterances that are processed by the CA.

This conceptual method will dynamically generate the knowledge base for the CA, which in theory will be much larger and more in depth than the traditional method of scripting CA knowledge bases, furthermore save large amounts of time and effort in the creation of new conversational agents. This will also allow the knowledge base to absorb the experiences of many more industry experts, with relative ease and new knowledge can be added more frequently and easily compared to existing knowledge base creation approaches.

Universal RESTful web service

Creating a universal web service from the engine to make UMAIR platform independent and therefore accessible from any device capable of accessing the internet for example smart phones and tablet devices. This will make the UMAIR more extensible and flexible as an application. This can then be furthered by turning the UMAIR in to XaaS like service, by offering CaaS (Conversation as a Service). Where an API can be developed that can be used to carry the user utterance through a URI (uniform resource identifier)/URL call and a JSON (JavaScript Object Notation) or XML response with the CA response and other information is delivered back to the caller in order to be parsed on any platform. Making the client side CA application light weight and platform independent.

PM and calculating the similarity strength of a user utterance to knowledge base resources is a processor intensive task (Lin et al., 2009, Lin and Dyer, 2010) which on normal everyday computers and mobile devices takes a long time. Therefore, this approach will also allow the server where the engine is deployed to do all the processor intensive tasks such as pattern matching and similarity calculations meaning that the CA can feasibly be deployed on any device, as the host device will not have to do the processing work.

Knowledge base expansion

As the knowledge base of UMAIR is designed in a modular fashion, future work can entail the expansion of the knowledge base to cover more aspects of the NADRA services such as birth certificates and family certificates. Moreover, the knowledge base could be expanded to cover all aspects of government related customer services, making UMAIR a single point of access to handle all government department related queries such as tax, housing or social related customer service.

References

Abandah, G. A., Jamour, F. T. & Qaralleh, E. A. (2014) Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)***,** 1-17.

Abdul-Mageed, M. & Korayem, M. (2010) Automatic identification of subjectivity in morphologically rich languages: the case of Arabic. Proceedings of the 1st workshop on computational approaches to subjectivity and sentiment analysis (WASSA), Lisbon. 2-6.

Adeeba, F. & Hussain, S. (2011) Experiences in building the Urdu WordNet. *Asian Language Resources collocated with IJCNLP 2011***,** 31.

Agbo-Ajala, O., Adeyemi, T., Ayeni, J. & Makinde, O. (2014) A Prototype Expert System for Human Disease Diagnosis. *Computing, Information Systems, Development Informatics & Allied Research Journal,* 5

Ahmed, T. & Butt, M. (2011) Discovering semantic classes for Urdu NV complex predicates. Proceedings of the Ninth International Conference on Computational Semantics. Association for Computational Linguistics, 305-309.

Ahmed, T. & Hautli, A. (2011) A first approach towards an Urdu WordNet. *Linguistics and Literature Review,* 1**,** 1-14.

Akram, F., Han, H.-S. & Kim, T.-S. (2014) A P300-based brain computer interface system for words typing. *Computers in biology and medicine,* 45**,** 118-125.

Akram, M. & Hussain, S. (2010) Word segmentation for urdu OCR system. Proceedings of the 8th Workshop on Asian Language Resources, Beijing, China. 88-94.

Albert, W. & Tullis, T. (2013) *Measuring the user experience: collecting, analyzing, and presenting usability metrics*, Newnes.

Alexander, I. & Maiden, N. (2005) *Scenarios, stories, use cases: through the systems development life-cycle*, John Wiley & Sons.

Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L. & Stent, A. (2001) Toward conversational human-computer interaction. *AI magazine,* 22**,** 27.

Almarsoomi, F. A., O'shea, J. D., Bandar, Z. A. & Crockett, K. A. (2012) Arabic word semantic similarity. Proceedings of World Academy of Science, Engineering and Technology. World Academy of Science, Engineering and Technology.

Alobaidi, O. G., Crockett, K. A., O'shea, J. D. & Jarad, T. M. (2013) Abdullah: An Intelligent Arabic Conversational Tutoring System for Modern Islamic Education. Proceedings of the World Congress on Engineering.

Alsanad, S. A. (2014) *The promotion of sustainable construction practices in Kuwait.* Ph.D, University of Manchester.

Altmann, E. G., Pierrehumbert, J. B. & Motter, A. E. (2009) Beyond word frequency: Bursts, lulls, and scaling in the temporal distributions of words. *PLoS One,* 4**,** e7678.

Anwar, W., Wang, X. & Wang, X.-L. (2006) A Survey of Automatic Urdu language processing. Machine Learning and Cybernetics, 2006 International Conference on. IEEE, 4489-4494.

Apple. (2014) *Siri Overview* [Online]. Available: http://www.apple.com/ios/siri/

[Accessed November 2014.

Artstein, R., Gandhe, S., Gerten, J., Leuski, A. & Traum, D. (2009) Semi-formal evaluation of conversational characters. *Languages: From Formal to Natural.* Springer.

Babu, S., Schmugge, S., Barnes, T. & Hodges, L. F. (2006) “What would you like to talk about?” an evaluation of social conversations with a virtual receptionist. Intelligent Virtual Agents. Springer, 169-180.

Baddeley, A. D. (1999) *Essentials of human memory*, Psychology Press.

Bellegarda, J. R. (2014) Spoken Language Understanding for Natural Interaction: The Siri Experience. *Natural Interaction with Robots, Knowbots and Smartphones.* Springer.

Bender, E. M. (2009) Linguistically naïve!= language independent: Why NLP needs linguistic typology. Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous? Association for Computational Linguistics, 26-32.

Bhatti, Z., Ismaili, I. A., Soomro, W. J. & Hakro, D. N. (2014) Word Segmentation Model for Sindhi Text. *American Journal of Computing Research Repository,* 2**,** 1-7.

Bickmore, T. & Cassell, J. (2005) Social Dialongue with Embodied Conversational Agents. *Advances in natural multimodal dialogue systems.* Springer.

Bickmore, T. & Giorgino, T. (2006) Health dialog systems for patients and consumers. *Journal of Biomedical Informatics,* 39**,** 556-571.

Bögel, T. & Butt, M. (2013) Possessive Clitics and Ezafe in Urdu. *Morphosyntactic Categories and the Expression of Possession,* 199**,** 291.

Boisseleau, W., Serban, O. & Pauchet, A. (2014). Building a narrative conversational agent using a component-based architecture. *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems.* Paris, France: International Foundation for Autonomous Agents and Multiagent Systems.

Boulis, C. & Ostendorf, M. (2005) Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. Proc. of the International Workshop in Feature Selection in Data Mining. Citeseer, 9-16.

Brom, C. & Lukavský, J. (2009) Towards More Human-Like Episodic Memory for More Human-Like Agents. Intelligent Virtual Agents. Springer, 484-485.

Brooke, J. (1996) SUS-A quick and dirty usability scale. *Usability evaluation in industry,* 189**,** 194.

Buchanan, B. G. & Shortliffe, E. H. (1984) *Rule-based expert systems*, Addison-Wesley Reading, MA.

Burkard, R. E. & Cela, E. (1999) *Linear assignment problems and extensions*, Springer.

Butt, M. (1995) *The structure of complex predicates in Urdu*, Center for the Study of Language (CSLI).

Butt, M., King, T. H. & Ramchand, G. (1994a) *Theoretical perspectives on word order in South Asian languages*, Center for the Study of Language (CSLI).

Butt, M. J., King, T. H. & Ramchand, G. C. (1994b) *Theoretical perspectives on word order in South Asian languages*, Center for the Study of Language and Inf.

Carroll, J. M. (1995) *Scenario-based design: envisioning work and technology in system development*, John Wiley and sons.

Cassell, J. (2000a) *Embodied conversational agents*, MIT press.

Cassell, J. (2000b) Embodied conversational interface agents. *Commun. ACM,* 43**,** 70-78.

Cassell, J., Bickmore, T., Campbell, L., Vilhjalmsson, H. & Yan, H. (2001) More than just a pretty face: conversational protocols and the affordances of embodiment. *Knowledge-Based Systems,* 14**,** 55-64.

Choi, S., Choi, J., Yoo, S., Kim, H. & Lee, Y. (2014) Semantic concept-enriched dependence model for medical information retrieval. *Journal of biomedical informatics,* 47**,** 18-27.

Chowdhury, G. G. (2003) Natural language processing. *Annual review of information science and technology,* 37**,** 51-89.

Coakes, S. J. & Steed, L. (2001) SPSS Analysis without Anguish Version 10.0 for Windows.

Crockett, K., Bandar, Z., O'shea, J. & Mclean, D. 18-23 July 2010 (2010) Goal Orientated Conversational Agents — The rocky road to commercialization. Fuzzy Systems (FUZZ), 2010 IEEE International Conference on. 1-8.

Crockett, K., Bandar, Z., O'shea, J. & Mclean, D. (2009) Bullying and debt: Developing novel applications of dialogue systems. *Knowledge and Reasoning in Practical Dialogue Systems (IJCAI)***,** 1-9.

Crockett, K., James, O. S. & Bandar, Z. (2011) Goal orientated conversational agents: applications to benefit society. *Agent and Multi-Agent Systems: Technologies and Applications.* Springer.

Damerau, F. J. (1964) A technique for computer detection and correction of spelling errors. *Communications of the ACM,* 7**,** 171-176.

Dasgupta, D., Hernandez, G., Garrett, D., Vejandla, P. K., Kaushal, A., Yerneni, R. & Simien, J. (2008) A comparison of multiobjective evolutionary algorithms with informed initialization and kuhn-munkres algorithm for the sailor assignment problem. Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation. ACM, 2129-2134.

Derrick, D. C. & Ligon, G. S. (2014) The affective outcomes of using influence tactics in embodied conversational agents. *Computers in Human Behavior,* 33**,** 39-48.

Durrani, N. (2007) *Typology of word and automatic word Segmentation in Urdu text corpus.* Citeseer.

Durrani, N. & Hussain, S. (2010) Urdu word segmentation. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 528-536.

Engelmore, R. S. & Feigenbaum, E. (1993) Expert systems and artificial intelligence. *EXPERT SYSTEMS,* 100**,** 2.

Engineering, C. F. L. (2014). Urdu Most Frequently Used Ligatures List. Al-Khwarizmi Institute of Computer Science, University of Engineering and Technology Pakistan.

Etemad-Sajadi, R. (2014) The influence of a virtual agent on web-users' desire to visit the company: The case of restaurant's web site. *International Journal of Quality & Reliability Management,* 31**,** 419-434.

Farukh, A. & Vulchanova, M. (2014) Predictors of Reading in Urdu: Does Deep Orthography Have an Impact? *Dyslexia,* 20**,** 146-166.

Fenton, N. E. & Pfleeger, S. L. (1998) *Software metrics: a rigorous and practical approach,* PWS Publishing Co.

Foster, J. J. (2001) *Data Analysis Using SPSS for Windows Versions 8-10: A Beginner's Guide,* Sage.

Galán, J. L., Merino, S., Martınez, J. & De Aguilera, M. (2013) A numerical and an exact approaches for classifying the items of a questionnaire into different competences. *Proceedings of Applications of Computer Algebra ACA 2013. Málaga***,** 301.

Gliem, J. A. & Gliem, R. R. (2003) Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales. Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education.

Google. (2014) *How to use ok google.* [Online]. Available: https://support.google.com/websearch/answer/2940021?hl=en 2014].

Gordon, R. G., Jr. (2005). Ethnologue: Languages of the World, Fifteenth edition. Dallas, Tex.: SIL International. .

Grandey, A. A., Dickter, D. N. & Sin, H. P. (2004) The customer is not always right: Customer aggression and emotion regulation of service employees. *Journal of Organizational Behavior,* 25**,** 397-418.

Gravetter, F. J. & Wallnau, L. B. (2002) Essentials of statistics for the behavioural sciences. *USA: Wadsworth*.

Griol, D., Carbo, J. & Molina, J. M. (2013) A statistical simulation technique to develop and evaluate conversational agents. *AI Communications,* 26**,** 355-371.

Hardie, A. (2003) Developing a tagset for automated part-of-speech tagging in Urdu. Corpus Linguistics 2003.

Haruechaiyasak, C., Kongyoung, S. & Dailey, M. (2008) A comparative study on thai word segmentation approaches. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on. IEEE, 125-128.

Hassenzahl, M. (2008). User experience (UX): towards an experiential perspective on product quality. *Proceedings of the 20th International Conference of the Association Francophone d'Interaction Homme-Machine.* Metz, France: ACM.

Hassenzahl, M., Diefenbach, S. & Göritz, A. (2010) Needs, affect, and interactive products–Facets of user experience. *Interacting with computers,* 22**,** 353-362.

Hayashi, Y. (2013). Pedagogical conversational agents for supporting collaborative learning: effects of communication channels. *CHI '13 Extended Abstracts on Human Factors in Computing Systems.* Paris, France: ACM.

Hijjawi, M., Bandar, Z., Crockett, K. & Mclean, D. (2014) ArabChat: an Arabic Conversational Agent. Computer Science and Information Technology (CSIT), 2014 6th International Conference on. IEEE, 227-237.

Hone, K. S. & Graham, R. (2000) Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, 6**,** 287-303.

Hussain, S. & Afzal, M. (2001) Urdu computing standards: Urdu zabta takhti (uzt) 1.01. Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International. IEEE, 223-228.

Hussain, S. & Durrani, N. (2008). A Study on Collation of Languages from Developing Asia. Lahore: Center for Research in Urdu Language Processing and the National University of Computer and Emerging Sciences. Online: http://idl-bnc. idrc. ca/dspace/bitstream/10625/42566/1/129903. pdf (accessed 12January 2012).

Ieee (2000) IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE Std 1471-2000***,** i-23.

Iqbal, F., Latif, A., Kanwal, N. & Altaf, T. (2011) Conversion of urdu nastaliq to roman urdu using OCR. Interaction Sciences (ICIS), 2011 4th International Conference on. IEEE, 19-22.

Janarthanam, S., Lemon, O., Bartie, P., Dalmas, T., Dickinson, A., Liu, X., Mackaness, W. & Webber, B. (2013) Evaluating a city exploration dialogue system combining question-answering and pedestrian navigation. Proc. ACL.

Jawaid, B. & Ahmed, T. (2009) Hindi to Urdu conversion: beyond simple transliteration. Conference on Language and Technology.

Kachru, Y. (1990) Hindi-Urdu in The Major Languages of South Asia. *The Middle East and Africa, edited by Bernard Comrie*.

Kaleem, M., Crockett, K. A. & O'shea, J. D. (2014a). Development of UMAIR the Urdu Conversational Agent for Customer Service. *Proceedings of the World Congress on Engineering*

Kaleem, M., O'shea, J. D. & Crockett, K. A. (2014b) Word order variation and string similarity algorithm to reduce pattern scripting in pattern matching conversational agents. Computational Intelligence (UKCI), 2014 14th UK Workshop on. IEEE, 1-8.

Kasap, Z., Ben Moussa, M., Chaudhuri, P. & Magnenat-Thalmann, N. (2009) Making them remember—emotional virtual characters with memory. *Computer Graphics and Applications, IEEE,* 29**,** 20-29.

Kasap, Z. & Magnenat-Thalmann, N. (2012) Building long-term relationships with virtual and robotic characters: the role of remembering. *The Visual Computer,* 28**,** 87-97.

Kaufmann, T., Völker, S., Gunesch, L. & Kübler, A. (2012) Spelling is just a click away–a user-centered brain–computer interface including auto-calibration and predictive text entry. *Frontiers in neuroscience,* 6.

Keeling, K., Beatty, S., Mcgoldrick, P. & Macaulay, L. (2004) Face Value? Customer views of appropriate formats for embodied conversational agents (ECAs) in online retailing. System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on. IEEE, 10 pp.

Khan, K., Siddique, M., Aamir, M. & Khan, R. (2012) An Efficient Method for Urdu Language Text Search in Image Based Urdu Text. *International Journal of Computer Science Issues(IJCSI),* 9.

Khan, Q. H. & Buchanan, L. (2014) Word frequency of written Urdu. *The Mental Lexicon,* 9**,** 131-140.

Kinner, P. R. & Gray, C. D. (2000). SPSS for windows made simple: Release 10. East Sussex: Psychology Press.

Kitchenham, B. A. & Pfleeger, S. L. (2002) Principles of survey research: part 3: constructing a survey instrument. *SIGSOFT Softw. Eng. Notes,* 27**,** 20-24.

Kuhn, H. W. (1955) The Hungarian method for the assignment problem. *Naval research logistics quarterly,* 2**,** 83-97.

Kulms, P., Kopp, S. & Krämer, N. C. (2014) Let's Be Serious and Have a Laugh: Can Humor Support Cooperation with a Virtual Agent? Intelligent Virtual Agents. Springer, 250-259.

Latham, A., Crockett, K. & Mclean, D. (2014) An adaptation algorithm for an intelligent natural language tutoring system. *Computers & Education,* 71**,** 97-110.

Latham, A., Crockett, K. A. & Bandar, Z. (2010a) A Conversational Expert System Supporting Bullying and Harassment Policies. ICAART (1). 163-168.

Latham, A. M. (2011) *Personalising Learning with Dynamic Prediction and Adaptation to Learning Styles in a Conversational Intelligent Tutoring System.* Manchester Metropolitan University.

Latham, A. M., Crockett, K. A., Mclean, D. A., Edmonds, B. & O'shea, K. 18-23 July 2010 (2010b) Oscar: An intelligent conversational agent tutor to estimate learning styles. Fuzzy Systems (FUZZ), 2010 IEEE International Conference on. 1-8.

Laugwitz, B., Held, T. & Schrepp, M. (2008) *Construction and evaluation of a user experience questionnaire*, Springer.

Lee, K. S., Lee, S. & Kim, H. (2014) Quick and participatory: adopting users' designs to improve a mobile app. CHI'14 Extended Abstracts on Human Factors in Computing Systems. ACM, 869-872.

Lehal, G. S. (2010) A word segmentation system for handling space omission problem in Urdu script. 23rd International Conference on Computational Linguistics. 43.

Lester, J., Branting, K. & Mott, B. (2004) Conversational agents. *The Practical Handbook of Internet Computing*.

Levenshtein, V. I. (1966) Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady. 707-710.

Li, F. & Jagadish, H. (2014) Constructing an Interactive Natural Language Interface for Relational Databases⇤. *Proceedings of the VLDB Endowment,* 8.

Li, Y. & Liu, B. (2007) A Normalized Levenshtein Distance Metric. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 29**,** 1091-1095.

Lin, J. & Dyer, C. (2010) Data-intensive text processing with MapReduce. *Synthesis Lectures on Human Language Technologies,* 3**,** 1-177.

Lin, J., Fu, D. & Zhu, J. (2009) What is Cloud Computing? *IT as a Service,* 11**,** 10.

Lutfi, S. L., Fernández-Martínez, F., Lorenzo-Trueba, J., Barra-Chicote, R. & Montero, J. M. (2013) I Feel You: The Design and Evaluation of a Domotic Affect-Sensitive Spoken Conversational Agent. *Sensors,* 13**,** 10519-10538.

Mahar, J., Shaikh, H. & Memon, G. (2012) A Model for Sindhi Text Segmentation into Word Tokens. *History,* 3**,** 37,997.

Mairesse, F., Walker, M. A., Mehl, M. R. & Moore, R. K. (2007) Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *J. Artif. Intell. Res.(JAIR),* 30**,** 457-500.

Malatesta, L., Raouzaiou, A., Karpouzis, K. & Kollias, S. (2009) Towards modeling embodied conversational agent character profiles using appraisal theory predictions in expression synthesis. *Applied intelligence,* 30**,** 58-64.

Malik, M. A. (2005) Towards a Unicode Compatible Punjabi Character Set. 27th Internationalization and Unicode Conference, Berlin.

Malik, M. A., Boitet, C. & Bhattacharyya, P. (2010) Analysis of Noori Nast'aleeq for Major Pakistani Languages. 2nd Workshop on Spoken Language Technologies for Under-resourced Languages SLTU-2010, Penang, Malaysia.

Malik, M. G. (2006) Punjabi machine transliteration. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 1137-1144.

Malik, M. G., Boitet, C. & Bhattacharyya, P. (2008) Hindi Urdu machine transliteration using finite-state transducers. Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics, 537-544.

Marietto, M. D. G. B., De Aguiar, R. V., Barbosa, G. D. O., Botelho, W. T., Pimentel, E., França, R. D. S. & Da Silva, V. L. (2013) Artificial Intelligence MArkup Language: A Brief Tutorial. *arXiv:1307.3091*.

Martinez, F. F., Blázquez, J., Ferreiros, J., Barra, R., Macias-Guarasa, J. & Lucas-Cuesta, J. M. (2008) Evaluation of a spoken dialogue system for controlling a hifi audio system. Spoken Language Technology Workshop, 2008. SLT 2008. IEEE. IEEE, 137-140.

Maulsby, D., Greenberg, S. & Mander, R. (1993) Prototyping an intelligent agent through Wizard of Oz. Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems. ACM, 277-284.

Metzler, D. (2008) Beyond bags of words: effectively modeling dependence and features in information retrieval. ACM SIGIR Forum. ACM, 77-77.

Michie, D. & Sammut, C. (2001) Infochat Scripter's Manual. *ConvAgent Ltd., Manchester*.

Microsoft. (2014) *Cortana Overview* [Online]. Available: http://www.windowsphone.com/en-us/features-8-1 2014].

Mills-Tettey, G. A., Stentz, A. & Dias, M. B. (2007) The dynamic hungarian algorithm for the assignment problem with changing costs.

Moller, S., Engelbrecht, K.-P., Kuhnel, C., Wechsung, I. & Weiss, B. (2009) A taxonomy of quality of service and quality of experience of multimodal human-machine interaction. Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on. IEEE, 7-12.

Mora-Cortes, A., Manyakov, N. V., Chumerin, N. & Van Hulle, M. M. (2014) Language Model Applications to Spelling with Brain-Computer Interfaces. *Sensors,* 14**,** 5967-5993.

Morik, K. (1991) Underlying assumptions of knowledge acquisition and machine learning. *Knowledge Acquisition,* 3**,** 137-156.

Mukund, S., Ghosh, D. & Srihari, R. K. (2010) Using cross-lingual projections to generate semantic role labeled corpus for Urdu: a resource poor language. Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 797-805.

Munkres, J. (1957) Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics,* 5**,** 32-38.

Musen, M. A. (1993) *An overview of knowledge acquisition*, Springer.

Naim, C. M. (1999) Introductory Urdu. rev. *Chicago: South Asia Language & Area Center, University of Chicago*.

Naseem, T. & Hussain, S. (2007) A novel approach for ranking spelling error corrections for Urdu. *Language Resources and Evaluation,* 41**,** 117-128.

Naz, S., Hayat, K., Imran Razzak, M., Waqas Anwar, M., Madani, S. A. & Khan, S. U. (2014a) The optical character recognition of Urdu-like cursive scripts. *Pattern Recognition,* 47**,** 1229-1248.

Naz, S., Razzak, M., Hayat, K., Anwar, M. & Khan, S. (2014b) Challenges in Baseline Detection of Arabic Script Based Languages. *In:* CHEN, L., KAPOOR, S. & BHATIA, R. (eds.) *Intelligent Systems for Science and Information.* Springer International Publishing.

Nolan, S. & Heinzen, T. (2011) *Statistics for the Behavioral Sciences (Loose Leaf)*, Macmillan Higher Education.

Nunamaker, J. F., Derrick, D. C., Elkins, A. C., Burgoon, J. K. & Patton, M. W. (2011) Embodied conversational agent-based kiosk for automated interviewing. *Journal of Management Information Systems,* 28**,** 17-48.

O'shea, K., Bandar, Z. & Crockett, K. (2009) A semantic-based conversational agent framework. Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for. IEEE, 1-8.

O'shea, J., Bandar, Z. & Crockett, K. (2011) Systems Engineering and Conversational Agents. *In:* TOLK, A. & JAIN, L. (eds.) *Intelligence-Based Systems Engineering.* Springer Berlin Heidelberg.

O'shea, J., Bandar, Z., Crockett, K. & Mclean, D. (2008) A comparative study of two short text semantic similarity measures. *Agent and Multi-Agent Systems: Technologies and Applications.* Springer.

O'shea, K. (2011) *A novel semantic-based conversational agent framework.* PhD, Manchester Metropolitan University

O'shea, K. (2013) Natural language scripting within conversational agent design. *Applied Intelligence***,** 1-9.

O'shea, K., Bandar, Z. & Crockett, K. (2010) A conversational agent framework using semantic analysis. *International Journal of Intelligent Computing Research (IJICR),* 1.

O'shea, K., Crockett, K., Bandar, Z. & O'shea, J. (2014) Erratum to: An approach to conversational agent design using semantic sentence similarity. *Applied Intelligence,* 40**,** 199-199.

Pallant, J. (2004) *SPSS survival manual: version 12,* Open University Press.

Pallant, J. & Manual, S. S. (2010) A step by step guide to data analysis using the SPSS program. *SPSS survival manual 4th ed,* 494.

Pickard, M. D., Burns, M. B. & Moffitt, K. C. (2013) A theoretical justification for using embodied conversational agents to augment accounting-related interviews. *Journal of Information Systems*.

Pietquin, O. & Hastie, H. (2013) A survey on metrics for the evaluation of user simulations. *Knowledge Eng. Review,* 28**,** 59-73.

Rashid, R. & Latif, S. (2012) A Dictionary Based Urdu Word Segmentation Using Maximum Matching Algorithm for Space Omission Problem. Asian Language Processing (IALP), 2012 International Conference on. IEEE, 101-104.

Rauschenberger, M., Schrepp, M., Cota, M. P., Olschner, S. & Thomaschewski, J. (2013) Efficient measurement of the user experience of interactive products. How to use the user experience questionnaire (ueq). example: spanish language version. *IJIMAI,* 2**,** 39-45.

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

Raza, A. & Hussain, S. (2010) Automatic diacritization for urdu. Proceedings of the Conference on Language and Technology. 105-111.

Raza, G. (2011) Subcategorization Acquisition and Classes of Predication in Urdu.

Richards, D. & Bransky, K. (2014) ForgetMeNot: What and how users expect intelligent virtual agents to recall and forget personal conversational content. *International Journal of Human-Computer Studies,* 72**,** 460-476.

Ristad, E. S. & Yianilos, P. N. (1998) Learning string-edit distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 20**,** 522-532.

Rizvi, S. M. J. & Hussain, M. (2005) Modeling case marking system of Urdu-Hindi languages by using semantic information. Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on. IEEE, 85-90.

Roy, B. & Graham, T. N. (2008) Methods for evaluating software architecture: A survey. *School of Computing TR,* 545**,** 82.

Rubin, V. L., Chen, Y. & Thorimbert, L. M. (2010) Artificially intelligent conversational agents in libraries. *Library Hi Tech,* 28**,** 496-522.

Rzepka, R. & Araki, K. (2015) ELIZA Fifty Years Later: An Automatic Therapist Using Bottom-Up and Top-Down Approaches. *Machine Medical Ethics.* Springer.

Sammut, C. (2001) Managing context in a conversational agent. *Linkoping Electronic Articles in Computer & Information Science,* 3.

Sankoff, D. & Kruskal, J. B. (1983) Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. *Reading: Addison-Wesley Publication, 1983, edited by Sankoff, David; Kruskal, Joseph B.,* 1.

Sarfraz, H., Dilawari, A. & Hussain, S. (2011) Assessing Urdu Language Support on the Multilingual Web.

Sarfraz, H., Hussain, S., Bano, M. & Dilawari, A. (2010) Technology preparedness for disseminating flood relief and rehabilitation information to local stakeholders online: Lessons learnt while developing Punjab flood relief website in Urdu.

Schlögl, S., Milhorat, P., Chollet, G. & Boudy, J. (2014) Designing Language Technology Applications: A Wizard of Oz Driven Prototyping Framework. *EACL 2014*, 85.

Secer, A., Sonmez, A. C. & Aydin, H. (2011) Ontology mapping using bipartite graph. *Int. J. Phys. Sci,* 6**,** 4224-4244.

Shawar, B. A. & Atwell, E. (2002). A comparison between ALICE and Elizabeth chatbot systems. Technical report, School of Computing, University of Leeds.

Shimazu, H. (2002) ExpertClerk: A Conversational Case-Based Reasoning Tool forDeveloping Salesclerk Agents in E-Commerce Webshops. *Artificial Intelligence Review,* 18**,** 223-244.

Silvervarg, A. & Jönsson, A. (2011) Subjective and objective evaluation of conversational agents in learning environments for young teenagers. 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Barcelona, Spain.

Sinha, C. & Hyma, R. (2013) ICT s AND SOCIAL INCLUSION. *Connecting ICTs to Development***,** 91.

Skantze, G. & Hjalmarsson, A. (2013) Towards incremental speech generation in conversational systems. *Computer Speech & Language,* 27**,** 243-262.

Sriramesh, K., Wattegama, C. & Abo, F. J. (2007) The role of ICTs in risk communication in Asia Pacific.

Studer, R., Benjamins, V. R. & Fensel, D. (1998) Knowledge engineering: Principles and methods. *Data & Knowledge Engineering,* 25**,** 161-197.

Syed, A., Aslam, M. & Martinez-Enriquez, A. (2014) Associating targets with SentiUnits: a step forward in sentiment analysis of Urdu text. *Artificial Intelligence Review,* 41**,** 535-561.

Tegos, S., Demetriadis, S. & Tsiatsos, T. (2014) A configurable conversational agent to trigger students' productive dialogue: a Pilot Study in the CALL domain. *International Journal of Artificial Intelligence in Education,* 24**,** 62-91.

Turing, A. M. (1950) Computing machinery and intelligence. *Mind***,** 433-460.

Turunen, M., Hakulinen, J. & Kainulainen, A. (2006) Evaluation of a spoken dialogue system with usability tests and long-term pilot studies: similarities and differences. INTERSPEECH.

Ultes, S., Schmitt, A. & Minker, W. (2013) On Quality Ratings for Spoken Dialogue Systems–Experts vs. Users. Proceedings of NAACL-HLT. 569-578.

Van Solingen, R., Basili, V., Caldiera, G. & Rombach, H. D. (2002) Goal Question Metric (GQM) Approach. *Encyclopedia of Software Engineering*.

Walker, M., Kamm, C. & Litman, D. (2000) Towards developing general models of usability with PARADISE. *Natural Language Engineering,* 6**,** 363-377.

Walker, M. A., Litman, D. J., Kamm, C. A. & Abella, A. (1997) PARADISE: A framework for evaluating spoken dialogue agents. Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 271-280.

Wallace, R. (2009) The Anatomy of A.L.I.C.E. *In:* EPSTEIN, R., ROBERTS, G. & BEBER, G. (eds.) *Parsing the Turing Test.* Springer Netherlands.

Weizenbaum, J. (1966) ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM,* 9**,** 36-45.

Whaley, L. J. (1996) *Introduction to typology: the unity and diversity of language*, Sage Publications.

Whaley, L. J. (1997) *Introduction to typology: The unity and diversity of language*, Sage.

Wilson, J. & Rosenberg, D. (1988) Rapid prototyping for user interface design.

Zafar, A., Mahmood, A., Abdullah, F., Zahid, S., Hussain, S. & Mustafa, A. (2012) Developing urdu wordnet using the merge approach. Proceedings of the Conference on Language and Technology. 55-59.

Zia, K. (1999) Towards Unicode Standard for Urdu. the Proceedings of Fouth Symposium on Multi-lingual Information Processing (MLIT4), Yangoon, Myanmar (CICC Japan).

Appendices

Appendix A – Questionnaire for UMAIR prototype one

MANCHESTER METROPOLITAN UNIVERSITY

UMAIR USABILIITY QUESTIONNAIRE

Dear Participant,

I am conducting a survey as part of my PhD research study. The purpose of this survey to determine and examine the usability, design and effectiveness of the UMAIR (Urdu Machine for Artificially Intelligent Recourse) Conversational Agent that you have just interacted with.

This questionnaire will take a maximum of 3 - 5 minutes to complete. Therefore I would appreciate it if you would take the time to complete it. The first questions require you to mark your answer on a scale in accordance to the question and the two possible responses. An answer of 1 on the scale would be strongly/very negative and an answer of 5 would be very positive.

Please be assured that individual confidentiality will be maintained and that only aggregate results will be reported. All collated data will be held for a maximum of 1 year, and then destroyed.

Thank you very much for your participation.

Participant Number

			1. Design			
Bad	①	②	③	④	⑤	Good
			2. Level of Helpfulness			
Low	①	②	③	④	⑤	High
			3. Quality of Information & Instructions			
Useless	①	②	③	④	⑤	Helpful
			4. Umair's Level of Understanding			
Low	①	②	③	④	⑤	High
			5. Conversation Level of Naturalness			
Low	①	②	③	④	⑤	High
			6. Your Satisfaction Level			
Low	①	②	③	④	⑤	High
			7. Time taken to complete the conversation			
Too Long	①	②	③	④	⑤	Acceptable
			8. Would you use this type of system again?			
		Yes		No		
			9. Would you use UMAIR instead of talking to a person face to face?			
		Yes		No		

Any further comments:

Appendix B - Questionnaire for UMAIR prototype two

☐ Participant Number

Gender: M/F Age: Country:

1. Design						
Bad	①	②	③	④	⑤	Very Good
2. Level of Helpfulness						
Bad	①	②	③	④	⑤	Very Good
3. Quality of Information & Instructions						
Bad	①	②	③	④	⑤	Very Good
4. Umair's Level of Understanding						
Bad	①	②	③	④	⑤	Very Good
5. Conversation Level of Naturalness						
Bad	①	②	③	④	⑤	Very Good
6. Your Satisfaction Level						
Bad	①	②	③	④	⑤	Very Good
7. Time taken to complete the conversation						
Bad	①	②	③	④	⑤	Very Good
8. Predictive text feature						
Bad	①	②	③	④	⑤	Very Good
9. Would you use this type of system again?						
		Yes		No		
10. Would you use UMAIR instead of talking to a person face to face?						
		Yes		No		

Any further comments:

Appendix C – UMAIR Log File

Con Num	User Utterance	UMAIR Response	Rule fired	Path followed?	Goal Achieved	Time	Strength
1	شناکتی کارڈ مجھے بنانا ہیں	کیا آپ کا تعلق پاکستان سے ہے ؟	4	N/A	FALSE	14:50:03	0.88
1	جی ہاں میں پاکستانی هوں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	14:50:44	1
1	نہیں یہ میرا سب سے پہلے ہے	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ١. نقل پرائش ,٢. مٹرک سنت , ٣. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	14:52:33	1
1	جی ہاں میرا پاس پیدائش کا سرٹیفکیٹ ہیں	کیا اپ نے فارم مکمل کر دیا ہ؟.	17	1	FALSE	14:53:11	1
1	جی ہاں فارم مکمل ہیں	کیا اپ نے دروست فیس درخواست کے ساتھ لگا دے ہے یا بینک میں جمع کروا دے ہے؟	18	1	FALSE	14:55:19	1
1	جی ہاں	اپ اپنی کارڈ درخواست قریبی نادرا دفترمیں کرا دیں	20	1	TRUE	14:55:59	1
2	سلام میرا کارڈ گم ہو گیا ہے	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ١. نقل پرائش ,٢. مٹرک سنت , ٣. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	N/A	FALSE	15:09:13	0.9444444
2	جی ہاں میرا پاس پیدائش کا سرٹیفکیٹ ہیں	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	15:10:22	1
2	نہیں	اپ کو فارم جمع کروانا ہو گاجو اس نیچے لنک سے اتار لیں یا قریبی نادرا دفتر سے حاصل کر لیں	19	1	TRUE	15:11:42	1
3	مجھے نیا شناختی کارڈ چاہینے	کیا آپ پاکستانی هیں؟	4	N/A	FALSE	15:26:55	0.8666667
3	جی نہیں میں انگلینڈ میں رہتے ہیں	کیا اپ کے والدین میں سے کوئی پاکستانی ہے؟	0	1	FALSE	15:27:40	1
3	بلکل نہیں میرے والدین میں سے کوئی نہیں	کیا اپ کی شادی کسی پاکستانی سے ہوئی ہے؟	12	1	FALSE	15:28:17	1
3	نہیں، میں اکیلی ہوں	میں معافی چاہتا ہوں، ان حالات میں یہ سب سے اچھا ہے کے اپ اپنے مقامی نادرا دفتر سے رابطہ کرے. وہاں جاؤ اور ہمارے ایجنٹوں سے بات کرے. وہ آپ کی مدد کر سکتے ہیں.	13	1	TRUE	15:29:43	1
4	شناکتی کارڈ مجھے بنانا ہیں	کیا آپ کا تعلق پاکستان سے ہے ؟	4	N/A	FALSE	17:05:29	1
4	جی نہیں میں انگلینڈ میں رہتے ہیں	کیا اپ کی ماں یا باپ پاکستانی ہے؟	0	1	FALSE	17:05:38	1
4	جی ہاں	آپ کے والدین کو ان کے پاکستانی شہریت کا کوئی ثبوت ہے؟	24	1	FALSE	17:06:04	1
4	ہاں جی	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	17:06:19	1
4	بلکل	کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	18	1	FALSE	17:06:29	1
4	بلکل	ٹھیک ہے اپ اپنی درخواست قریبی نادرا دفتر میں جما کروائیں	20	1	TRUE	17:06:44	1
5	میں ایک نیا شناختی کارڈ حاصل کرنے کے بارے پوچھنا چاہتا تھا	ٹیک شناکتی کارڈ حاصل کر نے کے لی لےا ، مجھے اپ کو تھوڑے سوال پوچھ نے پرن گے , کیا آپ پاکستانی هیں؟	4	N/A	FALSE	13:12:58	0.9583333
5	جي بلکل	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	13:13:22	1
5	جی ہاں میں نے پہلے ایک شناختی کارڈ ملا ہے	ٹیک تو اپ پرانا تجدید یا کھو کارڈ کو تبدیل کرنا چاہتے ہیں؟	6	1	FALSE	13:14:02	1
5	مجھے کارڈ تجدید کر نہ ہیں	اپ کے پاس پرانا شناختی کارڈ ہیں؟	7	1	FALSE	13:17:04	1
5	جي بلکل	آپ نے فارم مکمل کر لیا ہے؟	10	1	FALSE	13:17:34	1
5	جی ہاں	کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	18	1	FALSE	13:18:54	1

5	جی ہاں	ٹھیک ہے اپ اپنی درخواست قریبی نادرا دفتر میں جما کروائیں	20	1	TRUE	13:20:17	1
6	سلام شناکتی کارڈ مجھے بنانا ہیں	کیا آپ پاکستان کے شہری ہیں ؟	4	N/A	TRUE	13:31:07	0.88
6	جی ہاں میں پاکستانی ہوں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	13:31:34	1
6	نہیں میرا پہلا کارڈ نہیں بنا	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ١. نقل پرائش ,٢. مٹرک سنت , ٣. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	13:32:10	1
6	جی ہاں میرا پاس پیدائش کا سرٹیفکیٹ ہیں	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	13:32:59	1
6	جی بلکل	کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	18	1	FALSE	13:33:45	1
6	کیا آپ مجھے بتا سکتے ہیں فیس کتنی ہے	درست فیس اس لنک پر بیان کی گئی ہے (http://www.nadra.gov.pk/index.php/products/cards/cnic) آپ کی درخواست کی قسم (مثال کے طور پر، فاسٹ ٹریک، معیار) کے مطابق درست فیس کے ساتھ منسلک، اور پھر مکمل شدہ درخواست بھیجیں اپنے مقامی نادرا کے دفتر , لیکن, کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	29	3	FALSE	13:34:25	1
6	جی نہیں اب بر دوں گا	آپ کی درخواست کی قسم (مثال کے طور پر سٹینڈر یا فاسٹ ٹریک،) کے مطابق درست فیس کے ساتھ منسلک، اور پھر اپنے مقامی نادرا کے دفتر میں مکمل شدہ درخواست بھیجیں.	21	1	TRUE	13:35:48	1
7	سلام سر میرا شناختی کارڈ ختم ہو گیا ہے	اپ کے پاس پرانا شناختی کارڈ ہیں؟	7	N/A	FALSE	14:50:16	1
7	جی ہاں	بہت اچھا کیا اپ نے فارم بھر دیا ہیں؟	10	1	FALSE	14:50:36	1
7	مجھے کونسا فارم چاہئے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے, لیکن, آپ نے فارم مکمل کر لیا ہے؟	30	3	FALSE	14:51:15	1
7	جی نہیں	اپ کو فارم جمع کروانا ہو گاجو اس نیچے لنک سے اتار لیں یا قریبی نادرا دفتر سے حاصل کر لیں	19	1	TRUE	14:52:27	1
8	بھی کارڈ میرا گم ہو گیا ہے	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ١. نقل پرائش ,٢. مٹرک سنت , ٣. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	N/A	FALSE	15:05:33	1
8	ہاں جی پیدائشی سرٹیفکیٹ ہیں میرا پاس	کیا اپ نے فارم مکمل کر دیا ہ؟.	17	1	FALSE	15:06:18	1
8	نہیں	اپ نادرا کارڈ فارم اس لنک سے محفوظ کر سکتے ہیں یا نزدیکی دفتر سے حاصل کر سکتے ہیں	19	1	TRUE	15:06:45	1
9	جی مجھے کونسا فارم چاہئے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے	31	N/A	TRUE	15:52:19	1
10	میں صرف یہ جاننا چاہتے ہیں کے کونسا فارم چاہئے مجھے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے	31	N/A	TRUE	16:28:03	1
11	میں پوچھنا چاہتا ہوں کے کس طرح شناختی کارڈ بنایا جاتا ہے	کیا آپ پاکستان کے شہری ہیں ؟	4	N/A	FALSE	16:44:00	1
11	ہاں جی	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	16:44:47	1
11	نہیں میرا پہلا کارڈ نہیں بنا	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ١. نقل پرائش ,٢. مٹرک سنت , ٣. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	16:45:12	1
11	میرے ساتھ کوئی نہیں ہے ان مئی سے	معذرت ان دستاویزات کے بغیر اپ شناختی کارڈ نہیں بنوا سک تے. میں تم سے اس معاملے پر مزید مدد کے لئے اپنے مقامی نادرا دفتر کا دورہ مشورہ ہے.	16	1	TRUE	16:45:55	1
11	کیا آپ مجھے بتا سکتے نادرا مقامی دفتر ہے	اپ کا مقامی دفتر ١٢٣ گوجرہ روڈ، لاہور، پاکستان پر ہے.	27	N/A	TRUE	16:50:41	1
12	جی مجھے کونسا یہ پتا کر نہ ٹھا کے فیس کتنی لگتی ہے	درست فیس اس لنک پر بیان کی گئی ہے (http://www.nadra.gov.pk/index.php/products/cards/cnic) آپ کی درخواست کی قسم (مثال کے طور پر، فاسٹ ٹریک، معیار) کے مطابق درست فیس کے ساتھ منسلک، اور پھر مکمل شدہ درخواست بھیجیں اپنے مقامی نادرا کے دفتر	29	N/A	TRUE	17:02:12	1

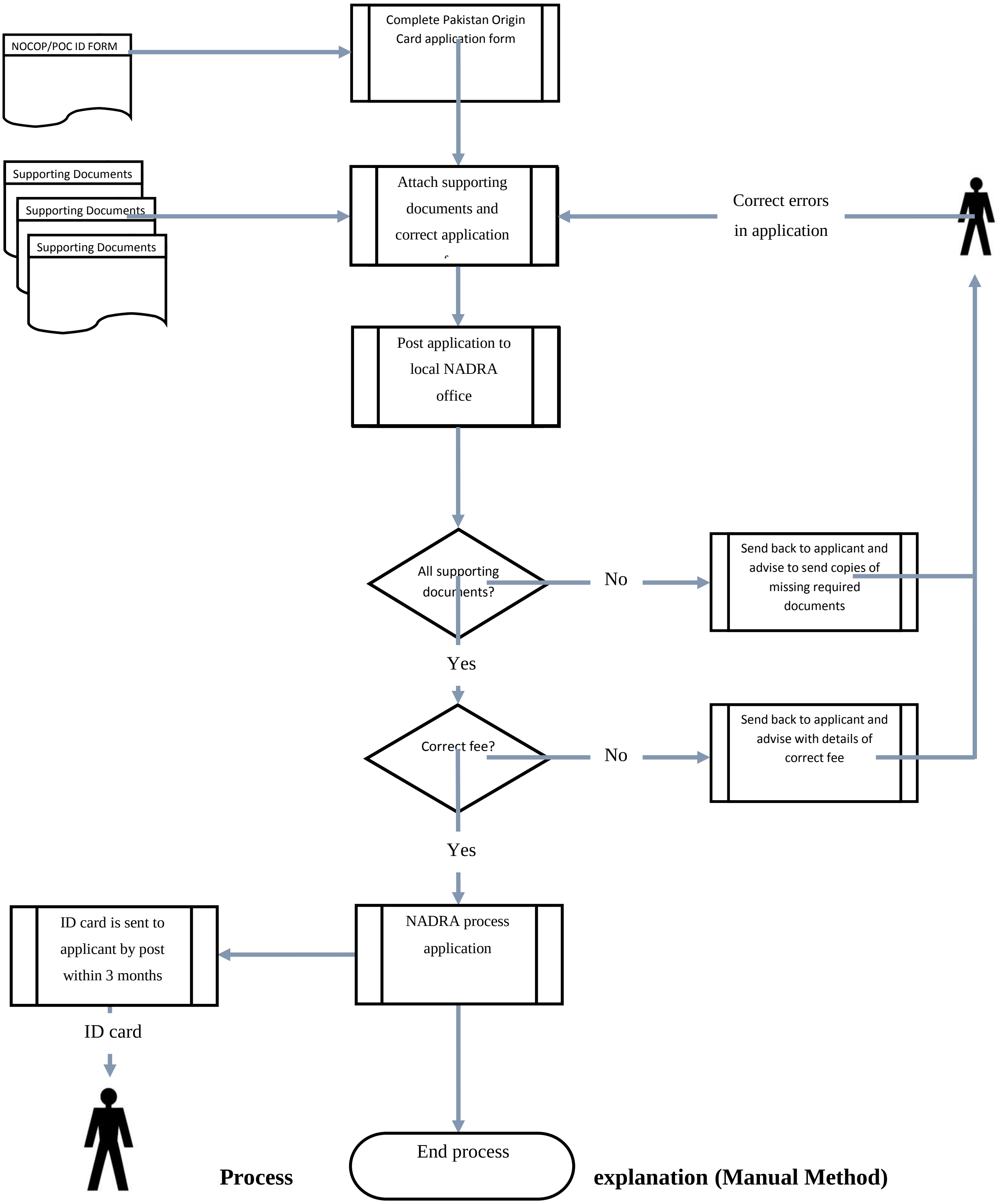
13	جی مجھے نیا شناکتی کارڈ بنانا ہیں	کیا آپ پاکستانی ھیں؟	4	N/A	FALSE	17:08:55	1
13	آپ پاگل ہیں	Warning	Bad Word	N/A	FALSE	17:10:01	0
13	نہیں	کیا اپ کے والدین میں سے کوئی پاکستانی ہے؟	0	1	FALSE	17:10:20	1
13	غلط	کیا اپ کسی پاکستانی سے شادی شدہ ہیں؟	12	1	FALSE	17:11:01	1
13	بلکل نہیں	میں معافی چاہتا ہوں، ان حالات میں یہ سب سے اچھا ہے کے اپ اپنے مقامی نادرا دفتر سے رابطہ کرے. وہاں جاؤ اور ہمارے ایجنٹوں سے بات کرے. وہ آپ کی مدد کر سکتے ہیں.	13	1	TRUE	17:11:21	1
14	شناکتی کارڈ مجھے بنانا ہیں	ٹیک شناکتی کارڈ حاصل کر نے کے لی لےا ، مجھے اپ کو تھوڑے سوال پوچھ نے پرن گے , کیا آپ پاکستانی ھیں؟	4	N/A	FALSE	17:31:03	0.88
14	ہاں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	17:32:07	1
14	نہیں	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	17:32:19	1
14	جی ہاں ہے	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	17:33:08	1
14	جی	اپ کو فارم جما کروانا ہو جا شناختی کارڈ کے لیجو کے اس لنک سے اتار سکتے ہیں یا قریب والے نادرا دفتر سے حاصل کر لیں .	19	1	TRUE	17:33:21	1
14	میں صرف پوچھنا چاہتا ہوں کے مکمل درخواست جہاں بھیجنے ہیں	اپنے مقامی نادرا دفتر کے مکمل شدہ درخواست بھیجیں	32	N/A	FALSE	17:34:17	1
14	اور کونسا فارم چاہئے مجھے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے	30	1	TRUE	17:34:55	1
15	میں پوچھنا چاہتی ہوں ایک شناختی کارڈ حاصل کرنے کے بارے میں	ٹیک شناکتی کارڈ حاصل کر نے کے لی لےا ، مجھے اپ کو تھوڑے سوال پوچھ نے پرن گے , کیا آپ پاکستانی ھیں؟	4	N/A	FALSE	15:16:17	1
15	جی ہاں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	15:16:33	1
15	جی ہاں	ٹیک تو اپ پرانا تجدید یا کھو کارڈ کو تبدیل کرنا چاہتے ہیں؟	6	1	FALSE	15:17:12	1
15	میں تجدید کر نہ ہیں کارڈ	اپ کے پاس پرانا شناختی کارڈ ہیں؟	7	1	FALSE	15:19:16	1
15	جي ہاں	بہت اچھا کیا اپ نے فارم بھر دیا ہیں؟	10	1	FALSE	15:19:59	1
15	مجھے فارم کونسا چاہئے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے	30	3	FALSE	15:20:45	1
15	جی ہاں مینے اس فارم کو بھر دیا ہے	کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	18	1	FALSE	15:22:13	1
15	بھائی مجھے بتاو فیس کتنی ہے	درست فیس اس لنک پر بیان کی گئی ہے (http://www.nadra.gov.pk/index.php/products/cards/cnic) آپ کی درخواست کی قسم (مثال کے طور پر، فاسٹ ٹریک، معیار) کے مطابق درست فیس کے ساتھ منسلک، اور پھر مکمل شدہ درخواست بھیجیں اپنے مقامی نادرا کے دفتر_, لیکن, کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	29	3	FALSE	15:23:34	1
15	جي ہاں میں نے ادا کر دیا ہے	ٹھیک ہے اپ اپنی درخواست قریبی نادرا دفتر میں جما کروائیں	20	1	TRUE	15:24:40	1
16	میں پوچھنا چاہتی ہوں ایک شناختی کارڈ حاصل کرنے کے بارے میں	ٹیک شناکتی کارڈ حاصل کر نے کے لی لےا ، مجھے اپ کو تھوڑے سوال پوچھ نے پرن گے , کیا آپ پاکستانی ھیں؟	4	N/A	FALSE	15:31:52	1
16	جی نہیں	کیا اپ کے والدین میں سے کوئی پاکستانی ہے؟	0	1	FALSE	15:32:13	1
16	نہیں میری ماں اور والد صاحب پاکستان نہیں ہیں	کیا اپ کی شادی کسی پاکستانی سے ہوئی ہے؟	12	1	FALSE	15:32:32	1
16	ہاں میرا شوہر پاکستانی ہے	آپ کے پاس وزارت داخلہ سے شہریت کا سرٹیفکیٹ ہے؟	14	1	FALSE	15:32:57	1

16	جی ہاں ہے	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	15:33:29	1
16	کونسا فارم چاہئے مجھے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے	30	3	FALSE	15:33:59	1
16	جی نہیں	اپ کو فارم جما کروانا ہو جا شناختی کارڈ کے لیجو کے اس لنک سے اتار سکتے ہیں یا قریب والے نادرا دفتر سے حاصل کر لیں .	19	1	TRUE	15:34:27	1
17	جی مجھے نیا شناکتی کارڈ بنانا ہیں	کیا آپ پاکستانی ھیں؟	4	N/A	FALSE	17:08:55	1
17	نہیں	کیا اپ کے والدین میں سے کوئی پاکستانی ہے؟	0	1	FALSE	17:10:20	1
17	نہیں	کیا اپ کسی پاکستانی سے شادی شدہ ہیں؟	12	1	FALSE	17:11:01	1
17	بلکل نہیں	میں معافی چاہتا ہوں، ان حالات میں یہ سب سے اچھا ہے کے اپ اپنے مقامی نادرا دفتر سے رابطہ کرے. وہاں جاؤ اور ہمارے ایجنٹوں سے بات کرے. وہ آپ کی مدد کر سکتے ہیں.	13	1	TRUE	17:11:21	1
18	سلام شناکتی کارد مجھے بنانا ہیں	کیا آپ پاکستان کے شہری ہیں ؟	4	N/A	TRUE	13:45:07	0.78
18	جی ہاں میں پاکستانی ھوں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	13:45:34	1
18	نہیں	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	13:46:10	1
18	ہاں میرا پاس پیدائش کا سرٹیفکیٹ ہیں	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	13:46:59	1
18	ہاں	کیا اپ نے درخواست کی دروست فیس جمع کروا دی ہے؟	18	1	FALSE	13:47:45	1
18	جی نہیں اب بر دوں گی	آپ کی درخواست کی قسم (مثال کے طور پر سٹینڈر یا فاسٹ ٹریک،) کے مطابق درست فیس کے ساتھ منسلک، اور پھر اپنے مقامی نادرا کے دفتر میں مکمل شدہ درخواست بھیجیں.	21	1	TRUE	13:45:48	1
19	شناکتی کارڈ مجھے بنانا ہیں	ٹیک شناکتی کارڈ حاصل کر نے کے لی لےا ، مجھے اپ کو تھوڑے سوال پوچھ نے پرن گے , کیا آپ پاکستانی ھیں؟	4	N/A	FALSE	17:31:03	0.88
19	ہاں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	17:32:07	1
19	نہیں	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	17:32:19	1
19	جی ہاں ہے	کیا اپ نے کارڈ درخواست کی دروست فارم ادا کر دے ہے؟	17	1	FALSE	17:33:08	1
19	جی	اپ کو فارم جما کروانا ہو جا شناختی کارڈ کے لیجو کے اس لنک سے اتار سکتے ہیں یا قریب والے نادرا دفتر سے حاصل کر لیں .	19	1	TRUE	17:33:21	1
20	میں صرف پوچھنا چاہتا ہوں کے مکمل درخواست جہاں بھیجنے ہیں	اپنے مقامی نادرا دفتر کے مکمل شدہ درخواست بھیجیں	32	N/A	FALSE	13:05:17	1
20	اور کونسا فارم چاہئے مجھے	اگر آپ ایک پاکستانی شہری ہیں تو اپ کو فارم کنیک چاہتے، لیکن اگر اپ بیرون ملک کی پاکستانی ہیں تو اپ کو نیکوپ فارم چاہتے	30	1	TRUE	13:05:55	1
21	میں ایک نیا شناختی کارڈ حاصل کرنے کے بارے پوچھنا چاہتا تھا	کیا آپ کا تعلق پاکستان سے ہے ؟	4	N/A	FALSE	16:46:23	1
21	عمیر آج موسم بہت خراب ہے	موسم مانچسٹر میں ہمیشہ برا ہے, لیکن, ٹیک شناکتی کارڈ حاصل کر نے کے لی لےا ، مجھے اپ کو تھوڑے سوال پوچھ نے پرن گے , کیا آپ پاکستانی ھیں؟	33	3	FALSE	16:46:40	1
21	ہاں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	16:47:33	1
21	نہیں	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	16:48:03	1
21	نہیں	معذرت ان دستاویزات کے بغیر اپ شناختی کارڈ نہیں بنوا سک تے. میں تم سے اس معاملے پر مزید مدد کے لئے اپنے مقامی نادرا دفتر کا دورہ مشورہ ہے.	16	1	TRUE	16:48:27	1

21	آپ پاگل ہیں	Warning	Bad Word	3	FALSE	16:48:43	0
21	آپ پاگل ہیں نا	Warning	Bad Word	3	FALSE	16:49:36	0
21	آپ بیوکوف ہیں	Warning	Bad Word	3	FALSE	16:50:15	0
21		Session terminated due to persistant use of bad language		N/A	FALSE	16:51:22	0
22	میں ایک نیا شناختی کارڈ حاصل کرنے کے بارے پوچھنا چاہتا تھا	کیا آپ کا تعلق پاکستان سے ہے ؟	4	N/A	FALSE	16:55:23	1
22	ہاں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	16:56:33	1
22	نہیں	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	16:57:03	1
22	جی نہیں، میری تمام ذاتی دستاویز ایک آگ میں کھو گئے تھے	معذرت ان دستاویزات کے بغیر اپ شناختی کارڈ نہیں بنوا سک تے. میں تم سے اس معاملے پر مزید مدد کے لئے اپنے مقامی نادرا دفتر کا دورہ مشورہ ہے.	16	1	TRUE	16:57:27	1
23	السلام وعلیکم کارڈ گم ہو گیا ہے میرا	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	N/A	FALSE	15:21:09	0.9444444
23	نہیں، میں ان میں سے کسی کے پاس ہے نہ	معذرت ان دستاویزات کے بغیر اپ شناختی کارڈ نہیں بنوا سک تے. میں تم سے اس معاملے پر مزید مدد کے لئے اپنے مقامی نادرا دفتر کا دورہ مشورہ ہے.	16	1	TRUE	15:22:29	1
24	میں ایک نیا شناختی کارڈ چاہتے ہیں، براہ مہربانی	کیا آپ کا تعلق پاکستان سے ہے ؟	4	N/A	FALSE	15:50:03	0.88
24	جی ہاں میں پاکستانی ہوں	کیا اپ نے کبی شناختی کارڈ بنوایا ہے؟	5	1	FALSE	15:50:44	1
24	نہیں یہ میرا سب سے پہلے ہے	کیا اپ کے پاسس مندرجہ ذیل کاغذات میں سے کوئی ایک ہے؟ ۱. نقل پرائش ,۲. مٹرک سنت , ۳. کسی خونی رشتے کا شناختی کارڈ , ٤. شہریت نامہ ٥. یا کوئی نہیں	15	1	FALSE	15:52:33	1
24	جی ہاں میرا پاس پیدائش کا سرٹیفکیٹ ہیں	کیا اپ نے فارم مکمل کر دیا ہ؟.	17	1	FALSE	15:53:11	1
24	جی ہاں فارم مکمل ہیں	کیا اپ نے دروست فیس درخواست کے ساتھ لگا دے ہے یا بینک میں جمع کروا دے ہے؟	18	1	FALSE	15:55:19	1
24	جی ہاں	اپ اپنی کارڈ درخواست قریبی نادرا دفترمیں کرا دیں	20	1	TRUE	15:55:59	1

Appendix D – Process flow charts

Process Title: Applying for a Pakistan National ID card



Process explanation (Manual Method)

Fill out the necessary application form relevant to the type of application (Overseas National or Pakistani National). Submit copies of the required documentation for the application. Include fee relevant to the type of application then submit the application form, required documents and fee to the local NADRA office, either by post or in person. The application is received and verified by the NADRA office, if there are any discrepancies in the application (documents, fees etc) the applications is sent back to the applicant and with instructions about what was wrong and how to correct it. If the application is correct it is processed by NADRA Islamabad, and the applicant normally receives the ID card applied for within 3 months.

Required documents for a new National Identity Card for Overseas Pakistanis (NICOP) and/or new Pakistan Origin Card (POC) (شناختی کارڈ)

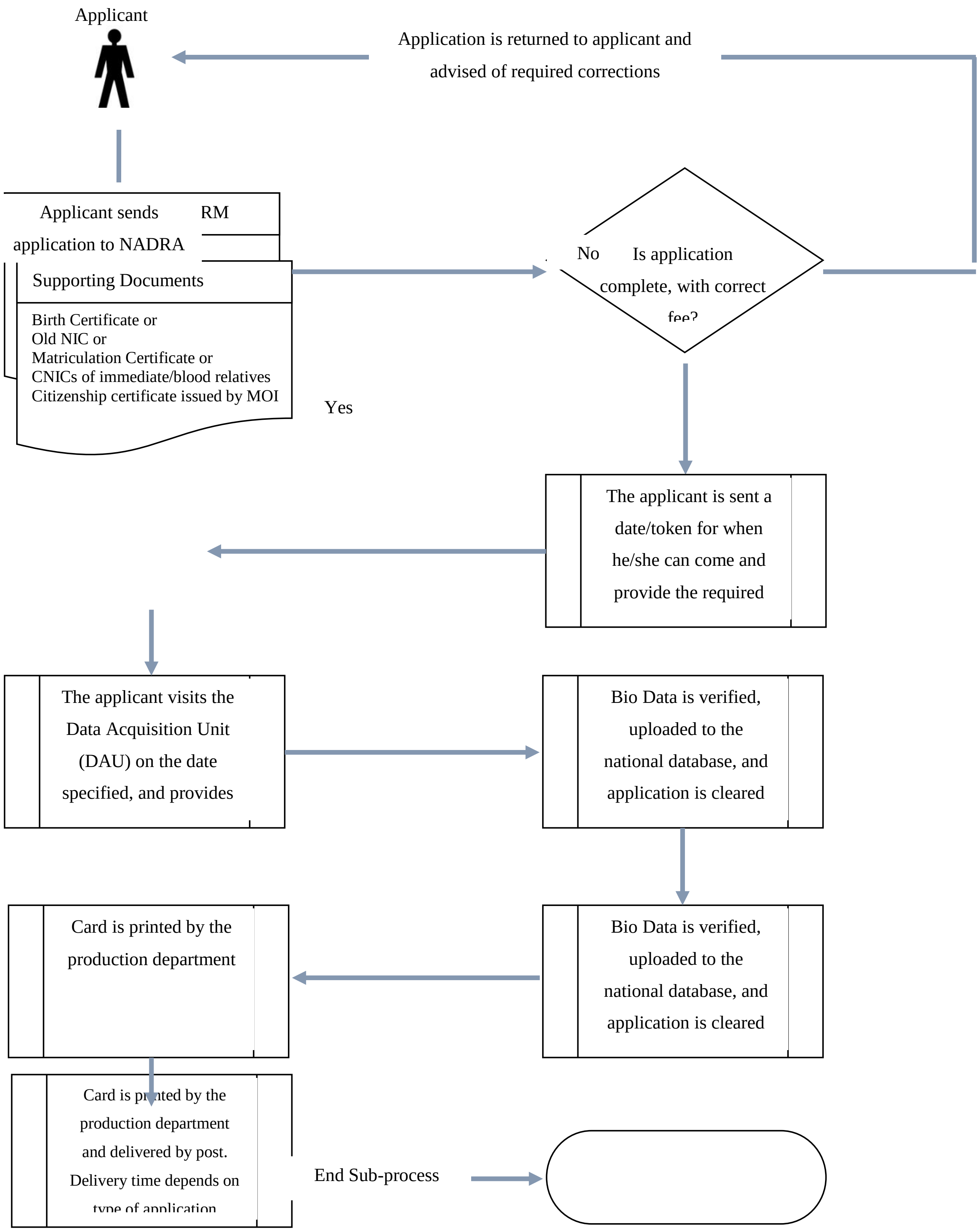
1. Old NIC (Manual ID card/Shanakhti Card) if available
2. Full birth certificate mentioning father/mother names (mandatory for Foreign / British born applicants and optional for applicants meeting criteria of Para -1 i.e. Old NIC).
3. CNIC/NICOP card of Father and Mother (Mandatory for Foreign / British born applicants and optional for applicants meeting criteria of Para -1 i.e. Old NIC).
4. CNIC/NICOP or its 13 digit number of Father / Mother or real Brother / Sister or Son / Daughter as reference for completion of family linkage (Mandatory for applicant applying on basis of para-1 i.e. old NIC).
5. Valid Pakistani Passport with valid visa or valid Foreign Passport (Mandatory for all applicants).
6. Nikah Nama / Marriage paper only for married applicants (optional for above 45 years age applicants).
7. Divorce paper / Death certificate of spouse in case of marital status Divorced / Widow.
8. Photocopies of all supporting documents. (NADRA team will not be able to process the application without photocopies of all required supporting documents).
9. Attestation of NICOP application form is mandatory and is responsibility of the applicant. Any Pakistani citizen holding valid NADRA Card can attest the application form except immediate family members.

Cost of Application

The details of the fees are as follows:

TYPE OF APPLICATION	**CARD TYPE**		
	NICOP	**POC**	
	All Applicants	Applicant 18 Years and above	Applicant under 18 Years
New Card	£40	£74	£41
Renewal / Duplicate / Modification	£46	£107	£60
Cancellation of NICOP / NIC / CNIC / POC	£67	£100	
New Card (Fast Track)	£64		
Renewal / Duplicate / Modification (Fast Track	£108		

Process Title: NADRA Services
Sub-process: Applying for a CNIC (Computerized National Identity Card)[1]



[1] CNIC (Computerized National Identity Card) is the core product of NADRA issued to a valid/legitimate citizen of Pakistan. It is a blend of state-of-the-art technology and well-defined business rules to guarantee its authenticity and validity. Every genuine, **18 Years and above**, citizen of Pakistan is eligible for CNIC.

Process Title: NADRA Services
Sub-process: Applying for a CNIC (Computerized National Identity Card)

NOTES

A CNIC applicant is required to produce the following documents at the time of application:

- Birth Certificate or
- Old NIC or
- Matriculation Certificate or
- CNICs of immediate/blood relatives
- Citizenship certificate issued by MOI

No documents are demanded from illiterate applicant for age verification for first time.

Residents of FATA/PATA will only be entertained at their native DAUs and their forms will be attested by concerned PA/APA.

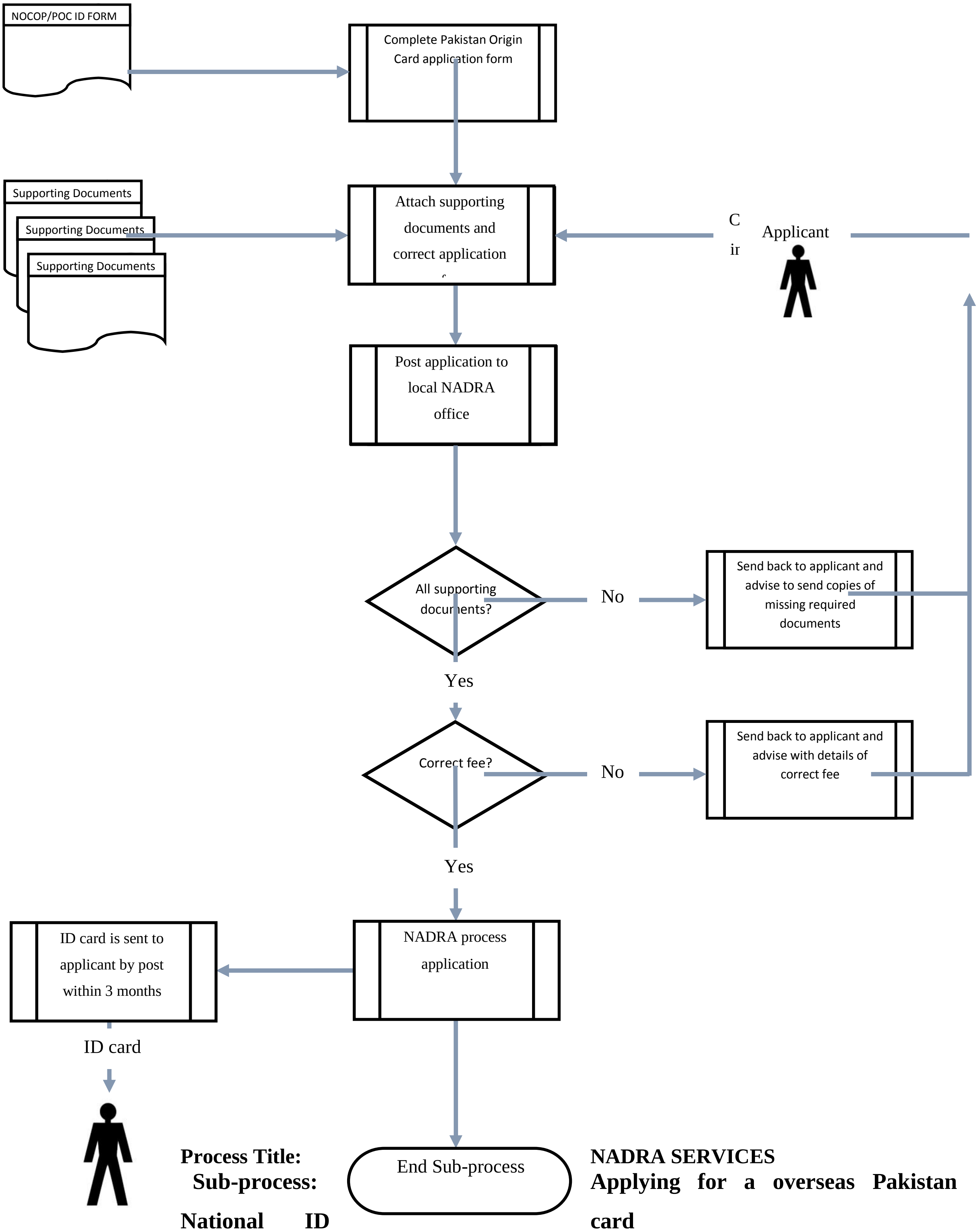
Fee Structure

Application Type	Normal	Urgent	Executive
SNIC		1500	
CNIC	200	300	
CRC	50	-	500
FRC	500	-	1000
Death Certificate		50	

Delivery Times

Application Type	Delivery Time
Normal	**30 days**
Urgent	**15 days**
Fast Track	**15 Days (Queue priority at NSRCs for immediate processing)**

Process Title: NADRA Services
Sub-process: Applying for a overseas Pakistan National ID card



Process Title: NADRA SERVICES
Sub-process: Applying for a overseas Pakistan National ID card

NOTES - Process explanation (Manual Method)

Fill out the necessary application form relevant to the type of application (Overseas National or Pakistani National). Submit copies of the required documentation for the application. Include fee relevant to the type of application then submit the application form, required documents and fee to the local NADRA office, either by post or in person. The application is received and verified by the NADRA office, if there are any discrepancies in the application (documents, fees etc) the applications is sent back to the applicant and with instructions about what was wrong and how to correct it. If the application is correct it is processed by NADRA Islamabad, and the applicant normally receives the ID card applied for within 3 months.

Required documents for a new National Identity Card for Overseas Pakistanis (NICOP) and/or new Pakistan Origin Card (POC) (شناختی کارڈ)

10. Old NIC (Manual ID card/Shanakhti Card) if available
11. Full birth certificate mentioning father/mother names (mandatory for Foreign / British born applicants and optional for applicants meeting criteria of Para -1 i.e. Old NIC).
12. CNIC/NICOP card of Father and Mother (Mandatory for Foreign / British born applicants and optional for applicants meeting criteria of Para -1 i.e. Old NIC).
13. CNIC/NICOP or its 13 digit number of Father / Mother or real Brother / Sister or Son / Daughter as reference for completion of family linkage (Mandatory for applicant applying on basis of para-1 i.e. old NIC).
14. Valid Pakistani Passport with valid visa or valid Foreign Passport (Mandatory for all applicants).
15. Nikah Nama / Marriage paper only for married applicants (optional for above 45 years age applicants).
16. Divorce paper / Death certificate of spouse in case of marital status Divorced / Widow.
17. Photocopies of all supporting documents. (NADRA team will not be able to process the application without photocopies of all required supporting documents).
18. Attestation of NICOP application form is mandatory and is responsibility of the applicant. Any Pakistani citizen holding valid NADRA Card can attest the application form except immediate family members.

Cost of Application

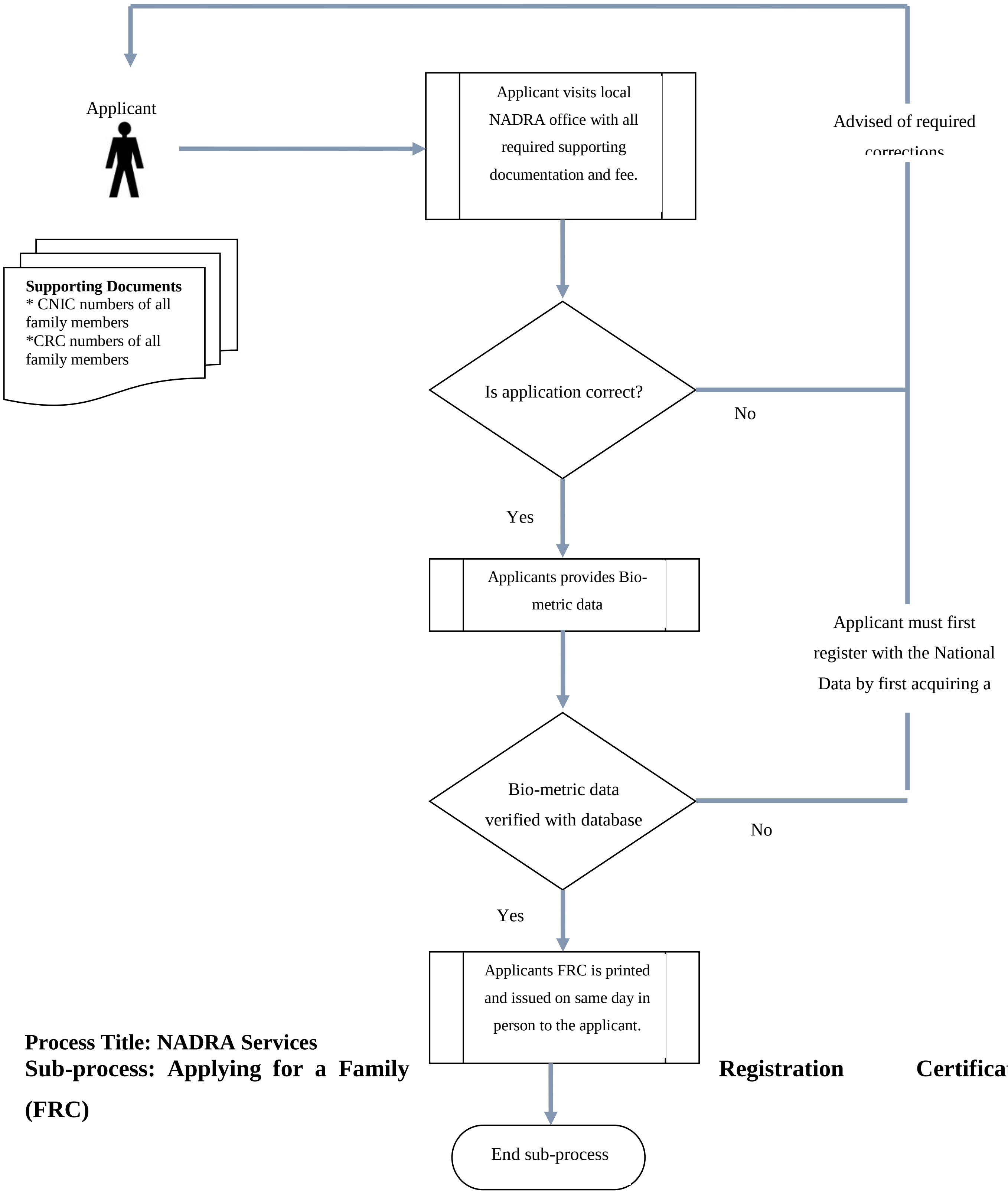
The details of the fees are as follows:

TYPE OF APPLICATION	CARD TYPE		
	NICOP	POC	
	All Applicants	Applicant 18 Years and above	Applicant under 18 Years
New Card	£40	£74	£41
Renewal / Duplicate / Modification	£46	£107	£60
Cancellation of NICOP / NIC / CNIC / POC	£67	£100	
New Card (Fast Track)	£64		
Renewal / Duplicate / Modification (Fast Track	£108		

Note: An additional £2 charge will apply on above-mentioned fees on all NICOP/POC applications submitted during NADRA Mobile Registration Team's Visits.

Process Title: NADRA Services

Sub-process: Applying for a Family Registration Certificate (FRC)[2]



Process Title: NADRA Services

Sub-process: Applying for a Family Registration Certificate (FRC)

[2] Family Registration Certificates are documents issued to nationals of Pakistan highlighting the family tree structure of the applicant. Certificate can be FRC by birth (Parents and siblings) or FRC by Marriage (Wife and children).

NOTES

Required Documents

- CNIC numbers of all family members
- CRC numbers of all family members

Delivery Time

Same day/Real time

Cost

Type	Normal (Rs.)
FRC	500

Appendix E – Evaluation scenarios for first and second evaluation

EVALUATION SCENARIOS UMAIR 1

Dear participant thank you for taking part in this study your scenario number is _____ read the instructions provided for your scenario number, and interact with the system to complete your task.

	DESCRIPTION	INSTRUCTIONS TO PARTICIPANT
1	New ID card required	You have never had an ID card and want to get a new one. You live in Pakistan and are over 18. Have all the necessary documentation requirements. Ask the system how to apply.
2	New ID card required	You want to renew an expired ID Card. You still have your old one.
3	New ID card form query	You don't know which form you are required to fill to get a new ID card. Ask the system that you want a new ID card and would like to just know which form is required.
99	New ID card required	You have never had an ID card and want to get a new one. You live in Pakistan and are over 18. You do not have any of the required documents.
4	New ID card price query	Ask the system the costs for a new ID card.
5	Lost ID card	You have lost your ID card and wish to get a new/replacement. You have a birth certificate. But have not filled in the form. Speak to the system to find out what you should do.
6	Lost ID card	You have lost your ID card and don't have the necessary documentation to get a new one. Speak to the system to find out what you should do.
7	Non – Pakistani national	You are a non Pakistan national, you have parents who are Pakistani nationals and you wish to apply for a new ID card. Speak to the system to find out what you should do.
8	Non – Pakistani national	You are a non Pakistan national, you are married to a Pakistani national and you wish to apply for a new ID card. You have a MOI certificate. Speak to the system to find out what you should do.
9	Non – Pakistani national	You are a non Pakistani national, not married to a Pakistani national, have not got parents who are Pakistani nationals, but wish to acquire an ID card. Speak to the system to find out what you should do.
10	Where to send application?	You are a non Pakistan national, you are married to a Pakistani national and you wish to apply for a new ID card. You do not have a MOI certificate. Speak to the system to find out what you should do.
11	Cost of fast track application?	Ask the system the cost of a fast track application.
12	How long does it take to get an ID card?	Ask the system how long it takes to get your ID card.
13	What time does the NADRA office close?	Ask the system of the opening hours of the NADRA office.
14	New ID card fee query	You would like to know what the fees are for applying for a new ID card.
15	Where is the NADRA office?	Find out where the NADRA office is, you live in Lahore.

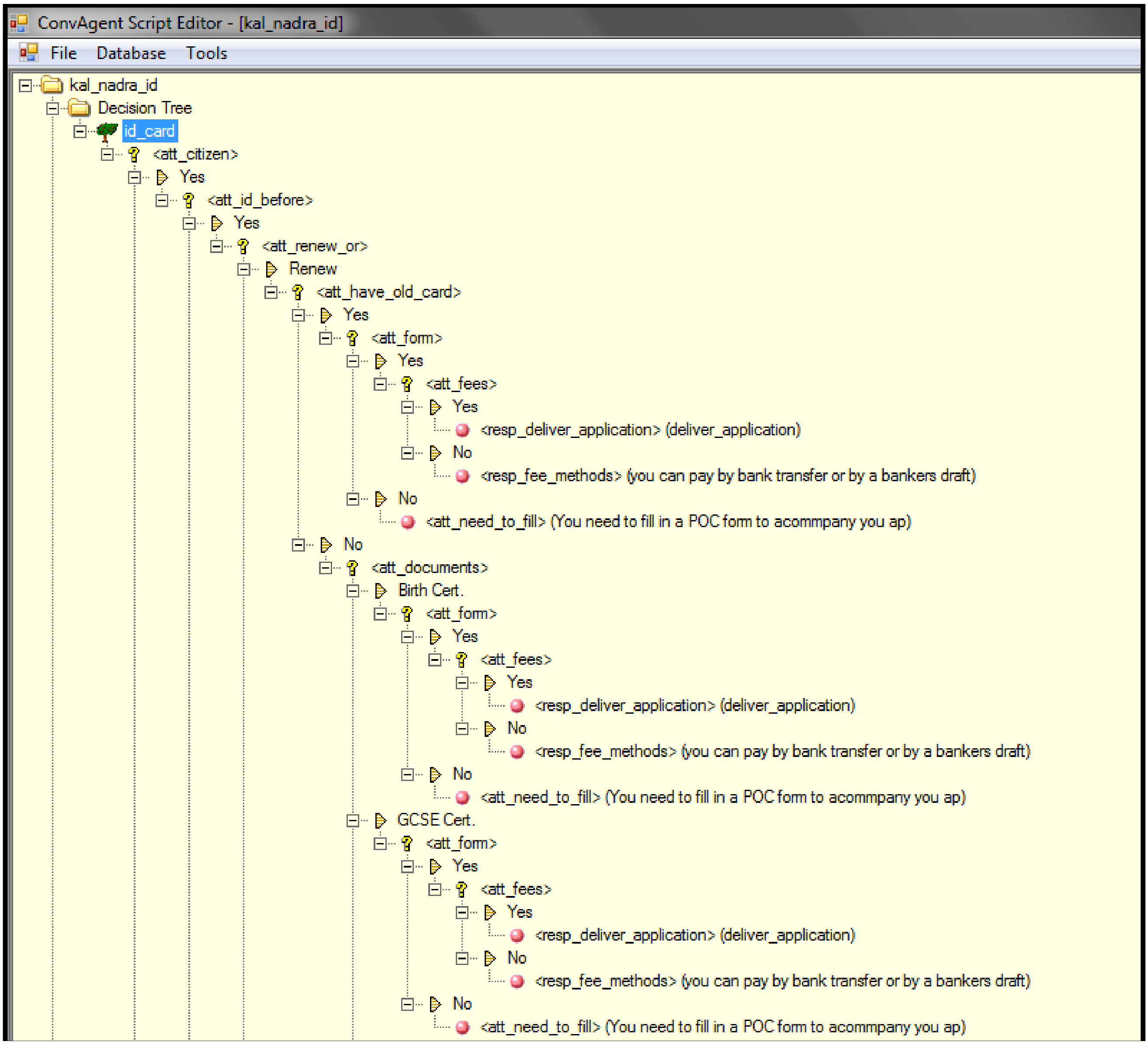
EVALUATION SCENARIOS UMAIR 2

Dear participant thank you for taking part in this study your scenario number is _____ read the instructions provided for your scenario number, and interact with the system to complete your task.

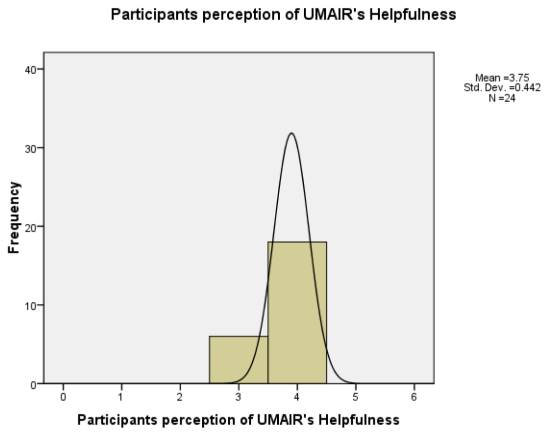
	DESCRIPTION	INSTRUCTIONS TO PARTICIPANT
1	New ID card required	You have never had an ID card and want to get a new one. You live in Pakistan and are over 18. Have all the necessary documentation requirements. Ask the system how to apply.
2	New ID card required	You want to renew an expired ID Card. You still have your old one.
3	New ID card form query	You don't know which form you are required to fill to get a new ID card. Ask the system that you want a new ID card and would like to just know which form is required.
4	New ID card required	You have never had an ID card and want to get a new one. You live in Pakistan and are over 18. You do not have any of the required documents.
5	New ID card price query	Ask the system the costs for a new ID card.
6	Lost ID card	You have lost your ID card and wish to get a new/replacement. You have a birth certificate. But have not filled in the form. Speak to the system to find out what you should do.
7	Lost ID card	You have lost your ID card and don't have the necessary documentation to get a new one. Speak to the system to find out what you should do.
8	Non – Pakistani national	You are a non Pakistan national, you have parents who are Pakistani nationals and you wish to apply for a new ID card. Speak to the system to find out what you should do.
9	Non – Pakistani national	You are a non Pakistan national, you are married to a Pakistani national and you wish to apply for a new ID card. You have a MOI certificate. Speak to the system to find out what you should do.
10	Non – Pakistani national	You are a non Pakistani national, not married to a Pakistani national, have not got parents who are Pakistani nationals, but wish to acquire an ID card. Speak to the system to find out what you should do.
11	Cost of fast track application?	Ask the system the cost of a fast track application.
12	How long does it take to get an ID card?	Ask the system how long it takes to get your ID card.
13	What time does the NADRA office close?	Ask the system of the opening hours of the NADRA office.

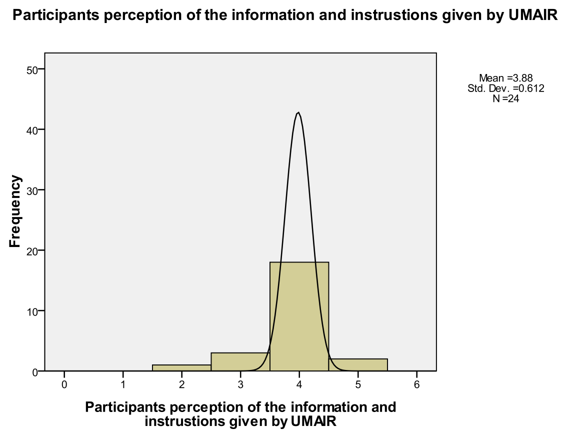
14	New ID card fee query	You would like to know what the fees are for applying for a new ID card.
15	Where is the NADRA office?	Find out where the NADRA office is, you live in Lahore.
16	New Passport	You would like to apply for a new passport, this is your first passport application and you have an ID card as proof of ID. Ask the system how to apply.
17	New Passport	You would like to apply for a new passport, it is you first passport you do not have any proof of ID. Ask the system how to apply.
18	Lost passport	You have lost your passport, you have proof of ID. Ask the system what you should do.
19	Which form for a new passport	Ask the system which form you are required to fill out in order to get a new passport.
20	How long for a new passport	Ask the system how long it take for a new passport application to be processed.
21	How much new Passport application	Ask the system how much it costs for a new passport.
22	Passport for an infant child	You have a new born child and wish to travel overseas, ask the system how to apply for a passport for an infant child.
23	Where to send application?	Ask the system where to send the completed passport application.
24	Which documents?	Ask the system which documents are required as proof of ID.

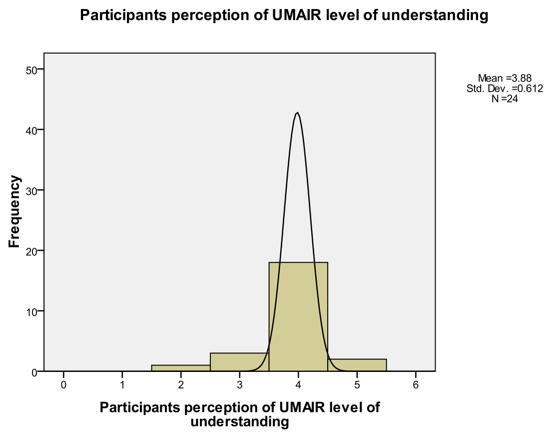
Appendix F – ConvAgent Tree Tool

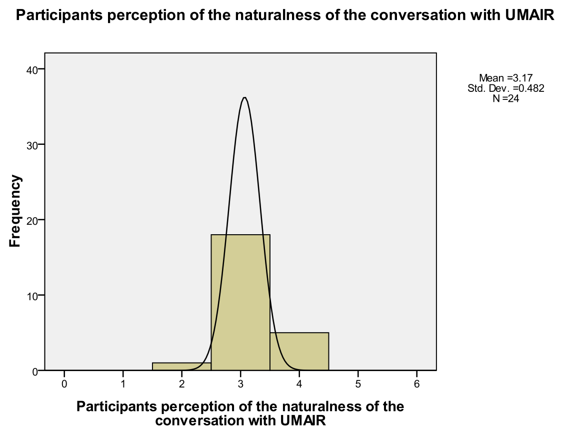

ConvAgent Script Editor - [kal_nadra_id]
File Database Tools
kal_nadra_id
Decision Tree
id_card
<att_citizen>
Yes
<att_id_before>
Yes
<att_renew_or>
Renew
<att_have_old_card>
Yes
<att_form>
Yes
<att_fees>
Yes
<resp_deliver_application> (deliver_application)
No
<resp_fee_methods> (you can pay by bank transfer or by a bankers draft)
No
<att_need_to_fill> (You need to fill in a POC form to acommpany you ap)
No
<att_documents>
Birth Cert.
<att_form>
Yes
<att_fees>
Yes
<resp_deliver_application> (deliver_application)
No
<resp_fee_methods> (you can pay by bank transfer or by a bankers draft)
No
<att_need_to_fill> (You need to fill in a POC form to acommpany you ap)
GCSE Cert.
<att_form>
Yes
<att_fees>
Yes
<resp_deliver_application> (deliver_application)
No
<resp_fee_methods> (you can pay by bank transfer or by a bankers draft)
No
<att_need_to_fill> (You need to fill in a POC form to acommpany you ap)

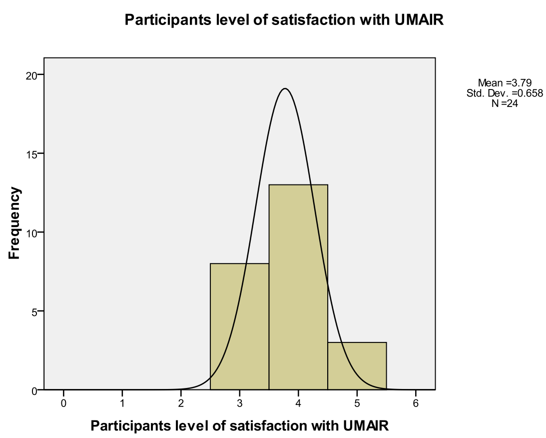
Appendix G – Normality Histograms first evaluation

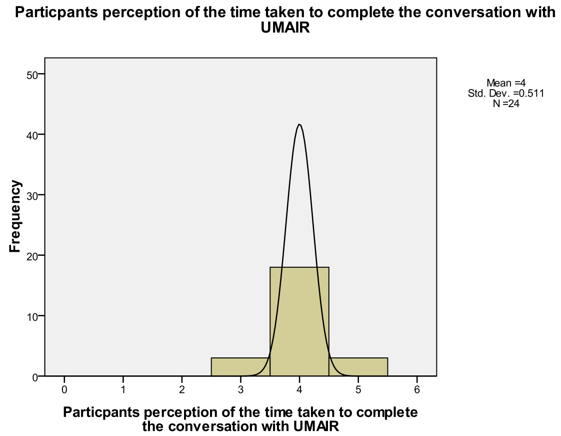

Participants perception of UMAIR's Helpfulness
Mean =3.75
Std. Dev. =0.442
N =24
Frequency
Participants perception of UMAIR's Helpfulness


Participants perception of the information and instrustions given by UMAIR
Mean =3.88
Std. Dev. =0.612
N =24
Frequency
Participants perception of the information and instrustions given by UMAIR

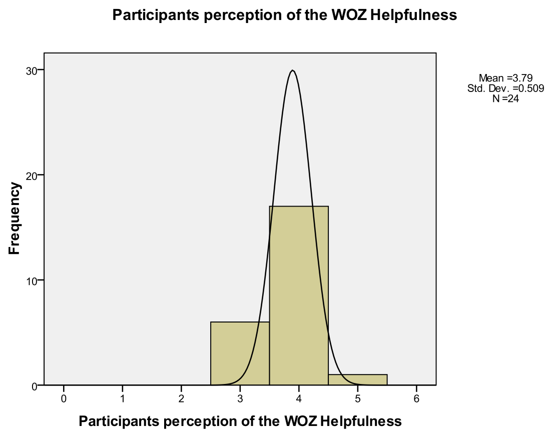

Participants perception of UMAIR level of understanding
Mean =3.88
Std. Dev. =0.612
N =24
Frequency
Participants perception of UMAIR level of understanding

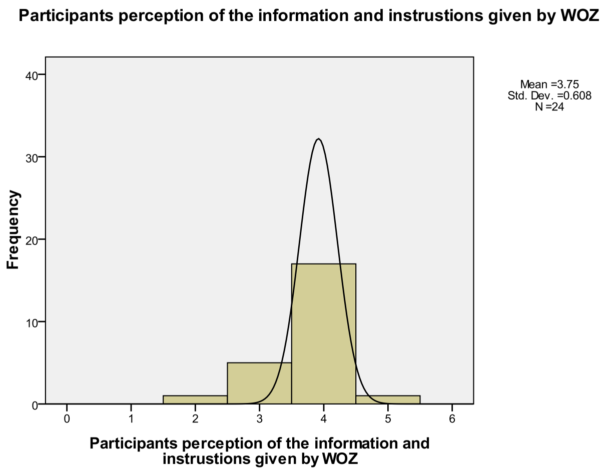

Participants perception of the naturalness of the conversation with UMAIR
Mean =3.17
Std. Dev. =0.482
N =24
Frequency
Participants perception of the naturalness of the conversation with UMAIR

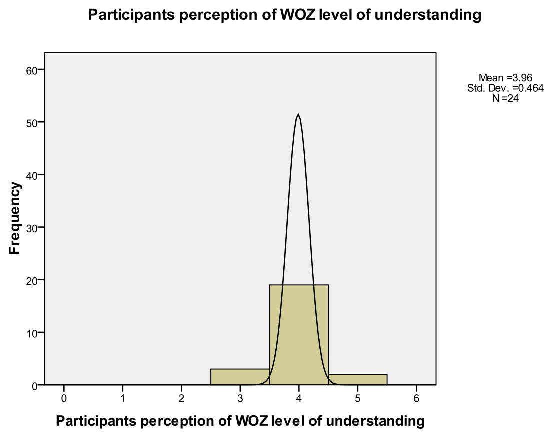

Participants level of satisfaction with UMAIR
Mean =3.79
Std. Dev. =0.658
N =24
Frequency
Participants level of satisfaction with UMAIR

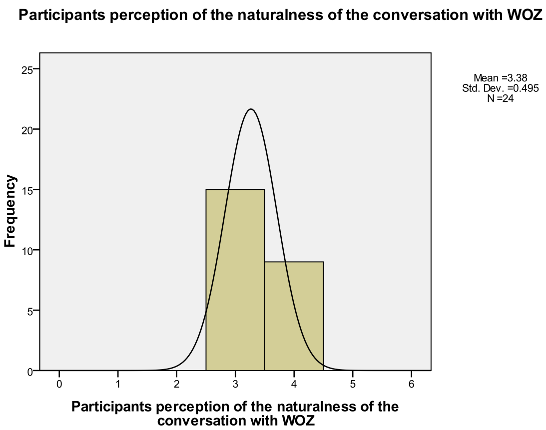

Particpants perception of the time taken to complete the conversation with UMAIR
Mean =4
Std. Dev. =0.511
N =24
Frequency
Particpants perception of the time taken to complete the conversation with UMAIR

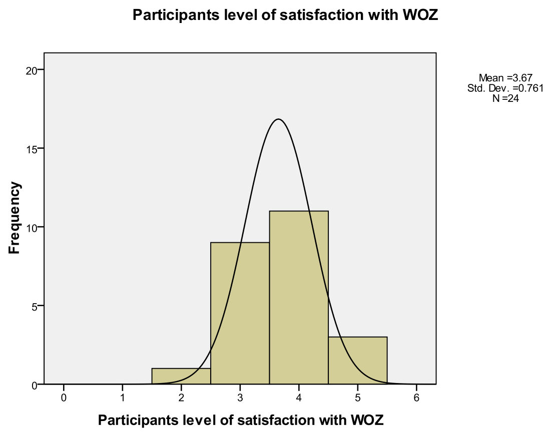
Continued

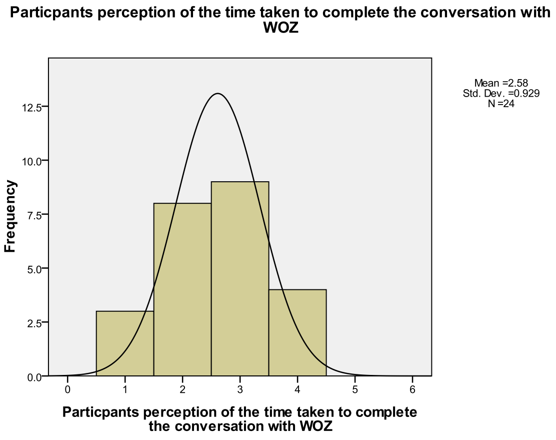

Participants perception of the WOZ Helpfulness
Mean =3.79
Std. Dev. =0.509
N =24
Frequency
0
10
20
30
0 1 2 3 4 5 6
Participants perception of the WOZ Helpfulness


Participants perception of the information and instrustions given by WOZ
Mean =3.75
Std. Dev. =0.608
N =24
Frequency
0
10
20
30
40
0 1 2 3 4 5 6
Participants perception of the information and instrustions given by WOZ

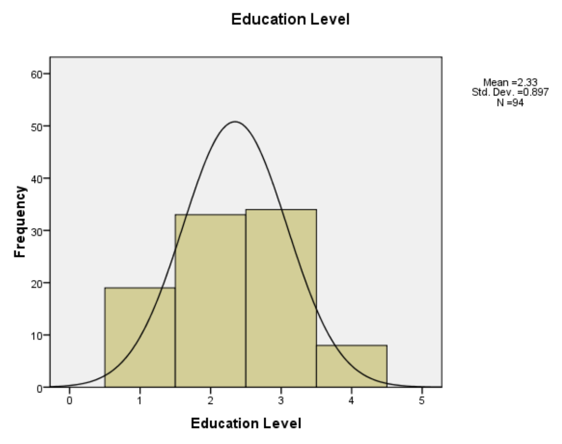

Participants perception of WOZ level of understanding
Mean =3.96
Std. Dev. =0.464
N =24
Frequency
0
10
20
30
40
50
60
0 1 2 3 4 5 6
Participants perception of WOZ level of understanding

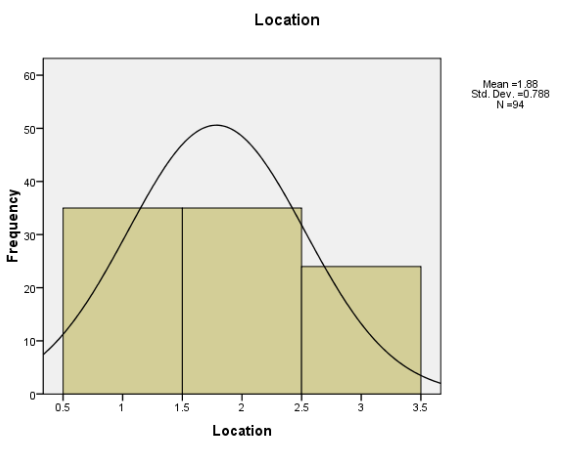

Participants perception of the naturalness of the conversation with WOZ
Mean =3.38
Std. Dev. =0.495
N =24
Frequency
0
5
10
15
20
25
0 1 2 3 4 5 6
Participants perception of the naturalness of the conversation with WOZ

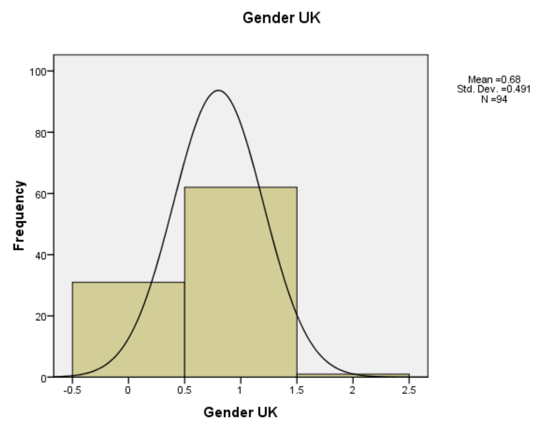

Participants level of satisfaction with WOZ
Mean =3.67
Std. Dev. =0.761
N =24
Frequency
0
5
10
15
20
0 1 2 3 4 5 6
Participants level of satisfaction with WOZ

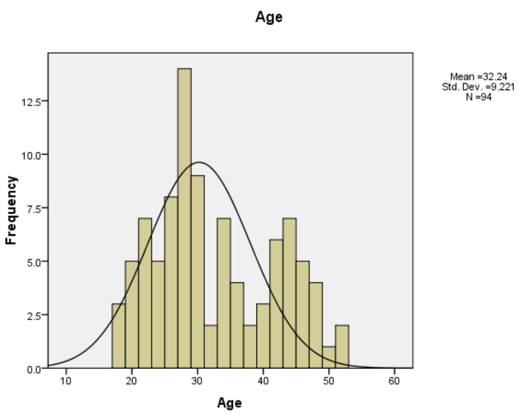

Particpants perception of the time taken to complete the conversation with WOZ
Mean =2.58
Std. Dev. =0.929
N =24
Frequency
0.0
2.5
5.0
7.5
10.0
12.5
0 1 2 3 4 5 6
Particpants perception of the time taken to complete the conversation with WOZ

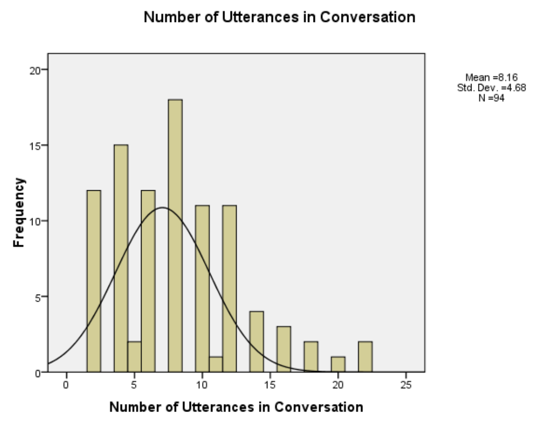
Appendix H – Normality Histograms second evaluation

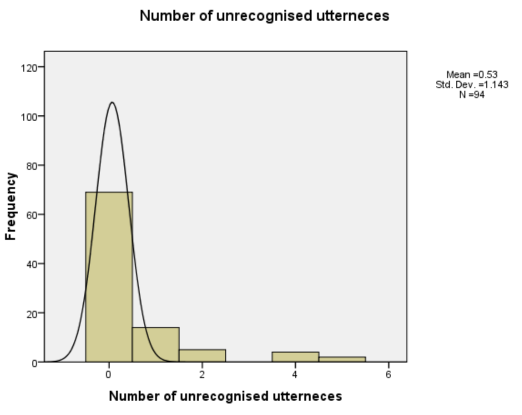

Education Level
Mean =2.33
Std. Dev. =0.897
N =94
Frequency
Education Level

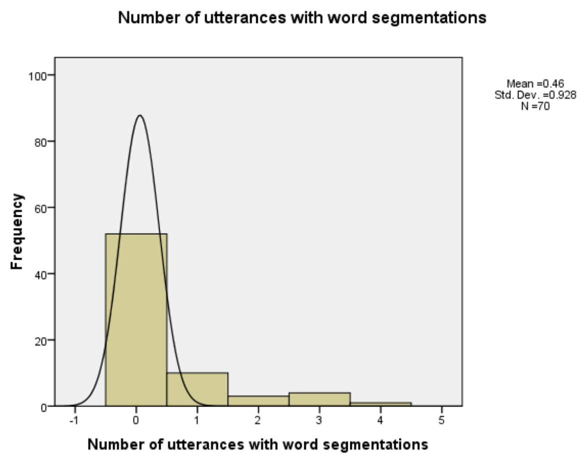

Location
Mean =1.88
Std. Dev. =0.788
N =94
Frequency
Location

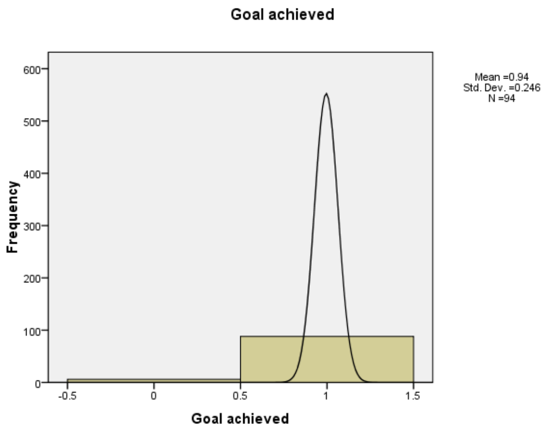

Gender UK
Mean =0.68
Std. Dev. =0.491
N =94
Frequency
Gender UK

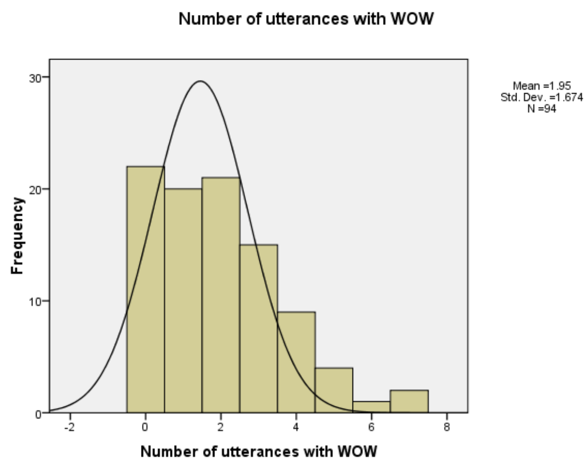

Age
Mean =32.24
Std. Dev. =9.221
N =94
Frequency
Age

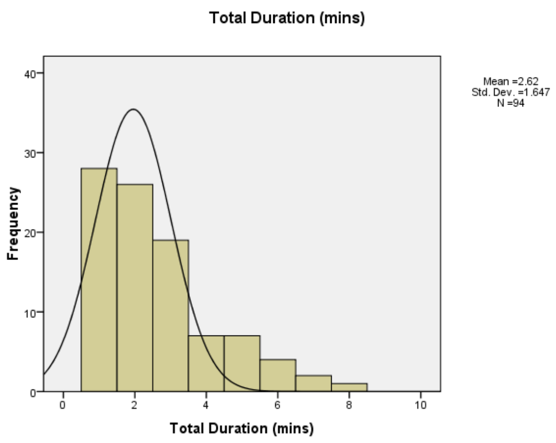

Number of Utterances in Conversation
Mean =8.16
Std. Dev. =4.68
N =94
Frequency
Number of Utterances in Conversation


Number of unrecognised utterneces
Mean =0.53
Std. Dev. =1.143
N =94
Frequency
Number of unrecognised utterneces


Number of utterances with word segmentations
Mean =0.46
Std. Dev. =0.928
N =70
Frequency
Number of utterances with word segmentations


Goal achieved
Mean =0.94
Std. Dev. =0.246
N =94
Frequency
Goal achieved


Number of utterances with WOW
Mean =1.95
Std. Dev. =1.674
N =94
Frequency
Number of utterances with WOW


Total Duration (mins)
Mean =2.62
Std. Dev. =1.647
N =94
Frequency
Total Duration (mins)

Appendix I - Industry contact approval

Mohammed Kaleem 06/03/2013

Dear Mr Iqbal, Thank you for taking the time to speak to me yesterday, furthe...

Kashif Iqbal/Project Manager/PMO/HQ <Kashif.Iqbal@nadra.gov.pk> 17/03/2013

to me

Dear Kaleem

I've gone through the document and I must say that you have understood our processes very well. Please go ahead and I wish you success in your efforts.

Good Luck!!

Regards
Kashif Iqbal
Deputy Director Projects

Sent from Samsung Mobile

...

Email: k4133m@gmail.com<mailto:k4133m@gmail.com>

Appendix J – Interview Questions

Thanks for taking the time to speak to me with regards to the NADRA processes and procedures. I would like to ask you some questions to clarify the internal procedures and processes for the purpose of my PhD research.

1. Can you please explain to me the process a typical applicant has to go through in order to get an ID and passport (i.e. application process, forms to be filled, documents required form the applicant, fees, where the applicant submit their applications etc.)?
2. In a case where the applicant doesn't have the necessary ID what is the procedure/instructions given to them by the customer service representative?
3. What advice is given by a customer adviser in a case where they cannot deal/help the applicant, due to circumstances where the applicant is missing documents etc.?
4. What do overseas applicants need to obtain a Pakistani ID card?
5. How do foreign nationals who are married to Pakistani nationals apply for ID cards/passports?
6. How do the NARDA staff deal with exceptional cases? (for example customers with no proof of ID or overseas nationals with no family or spousal proof of nationality)
7. What are the opening times for the NARDA offices?
8. How do the NARDA customer service staff deal with abusive customers?
9. Are there helplines/websites available for the customers to find detailed information from?
10. Are there any helplines customers can call?
11. What are the most frequently asked questions the customer services staff face?

Appendix K - Author Publications

Development of UMAIR the Urdu Conversational Agent for Customer Service

M. Kaleem[3], J. O'Shea[3] and K. Crockett[3]

Abstract—**This paper outlines the development of UMAIR an Urdu conversational agent developed as a customer service representative. UMAIRs architecture includes a novel engine, scripting language and WOW (Word Order Wizard) string similarity algorithm which are combined to tackle the language unique challenges of Urdu. Initial testing of the new architecture has yielded positive results towards UMAIR being able to cope with the inherent differences in the Urdu language such as word order.**

Index Terms—**Conversational Agents, Dialog Systems, Sentence Similarity, Urdu**

INTRODUCTION

Conversational Agents (CAs) essentially allow people to interact with computer systems intuitively using natural language dialogue [1]. In today's increasingly complex business environment, organisations face pressures regarding cost reduction, engagement scope, and attention to quality [2]. With this in mind, one of the most important emerging applications of CAs is online customer self-service/assistance, providing the user with the kind of services that would come from a knowledgeable or experienced human [3]. Following several years of research and development activities, CAs in English, European and East Asian languages CAs have become a popular area. However, South Asian Languages especially Urdu have received less attention [4]. Urdu is the national language of Pakistan, one of the state languages of India, has more than 60 million first language speakers and more than 100 million total speakers in more than 20 countries [5]. Urdu script is written from right to left like the Semitic languages having a morphology similar to Arabic, Persian and Pashto language letters [6].

In 2008 Pakistan was hit by the worst floods in its history, in light of this natural disaster a relief website was set up in English to disseminate vital information about help, rescue efforts and shelter to those affected and displaced by the floods. However, the website proved to be quite ineffective until it was translated into Urdu. Hussain, [7] states that traditionally ICT solutions have been deployed in the English language, but it is evident that in order to reach the masses, the language medium needs to be one that is understood by the masses. Inevitably the web is playing a pivotal role in bringing information to the populations around the world [8]. Information available in localized contexts is more relevant to speakers of different languages; this is one of the drivers of this research.

It is made apparent that there is a genuine necessity for CA research in Urdu to facilitate better access to information to the mass population while taking advantage of the unique features CAs can provide.

This motivated the research and development of a prototype CA named UMAIR (Urdu Machine for Artificial Intelligent Recourse) which was developed initially to answer customer/user queries on the domain of ID card application in Pakistan. One of the main challenges that came with the Urdu language was that Urdu does not have the computational lexical resources that are readily available to western languages such as WordNet [9]. There have been several factors causing slow growth of Urdu software. One factor has been the lack of standards for Urdu computing [10]. Ahmed and Butt [11] argue that one of the major bottlenecks for Urdu software development is the lack of lexical resources available for the Urdu language, for example the Urdu language doesn't have the established electronic infrastructures that are taken for granted in English and other European languages.

Consequently the research and development of an Urdu Conversational Agent is not simply a matter of re-engineering existing methods and algorithms. Novel CA engine components need to be researched and developed capable of handling the inherent differences in the Urdu language. Traditionally Conversational agents use a Pattern Matching (PM) technique to match user utterances to a repository of scripted pre-anticipated utterances and their appropriate responses. Over the years this method although reliable, has proven to be a laborious and time consuming task.

This paper is organized as follows: Section II provides an overview of conversational agents and their areas of application. Section III and IV present a summary of the Urdu language and outline the challenges Urdu poses to the implementation of a novel Urdu conversational agent. Section V details the process of knowledge engineering for the domain. Section VI and VII introduce UMAIR and the components that make up the architecture. Sections VIII, IX and X detail the evaluation methodology, the results and conclusions that derived from them.

CONVERSATIONAL AGENTS

CA Background

The term "Conversational Agent" is interpreted in various ways by different researchers; Chen [12], defines them as a natural language interaction interface designed to simulate conversation with a real person. Cohen [13] describe CAs as an agent which uses natural language

[3] The Intelligent Systems Group, Department of Computing Mathematics & Digital Technology, Manchester Metropolitan University, Manchester, M1 5GD, UK. Email: k4133m@gmail.com (j.d.oshea, k.crockett)@mmu.ac.uk

dialogue to communicate with users. Nevertheless the essence of CAs which is agreed upon is that natural language dialogue is utilized between the human and an application running on a computer [1]. There are two main types of CAs Goal Orientated CAs (GO-CA) and General CAs. GO-CAs direct the user's discussion towards a goal e.g. getting some information or help. Whereas a general CA's goal is to just continue the conversation. Conversational agents are representative intelligent agents that are able to respond to user requests and queries in an intelligent way (with natural language dialogue). They can understand the intention of users through conversation, normally through a text based interface. A CA also has the ability to reason and pursue a course of action based on its interactions with humans and other agents [14].

One of the earliest CAs developed was ELIZA [15]. ELIZA was a Chatbot capable of creating the illusion that the agent was actually listening and understanding the user's utterances and providing intelligent response, however it was just using simple pattern matching techniques that worked by simply parsing and recomposing key words based on the user input to formulate responses. As the field of CA's advanced, ALICE (Artificial Linguistic Intelligent Computer Entity) was produced. The knowledge base for ALICE is stored in AIML (Artificial Intelligent Markup Language) files. Fundamentally AMIL is in essence a PM scripting language derived from Extensible Markup Language (XML) and used symbolic reduction to parse user utterances and generate responses. In ALICE, the AIML technology was responsible for pattern matching and to relate a user input with a response in the chatterbot's Knowledge Base (KB) [16]. In essence the ALICE engine was a more refined version of the simpler engine used in ELIZA [17] but still lacked the sophistication of more recent engines. An example of a more recent CA is InfoChat [18]. InfoChat implements a pattern matching approach using a sophisticated scripting language known as Pattern Script. InfoChat scripting language is a rule-based language, which depends on a rule based structure to handle the expected conversation, However, it also uses the concept of "spreading activation", which strengthens or inhibits rule firing based on conversation history. The similarity is calculated through several parameters such as activation level and pattern strength.

How do CAs work?

CAs have been developed using many different techniques. The three main techniques are Natural Language Processing (NLP) and Short Text Semantic Similarity (STSS) and Pattern Matching (PM). NLP is an area of research that explores how computers can be used to understand and manipulate natural language text or speech to do useful things [19]. NLP assumes certain aspects for it to work effectively. The utterance is expected to be grammatically correct which usually it is no, incorrect sentences may be "repaired" but this add computational overhead. Another point is that languages are very rich in form and structure, and contain ambiguities. A word might have more than one meaning (lexical ambiguity) or a sentence might have more than one structure (syntactic ambiguity/free word order), in light of this the NLP approach is not suitable to develop a CA in the Urdu language. Another approach that is adopted in the development of CAs is the utilization of STSS measures to gauge the similarity between short sentences (10 – 25 words longs) [3]. Through employing sentence similarity measures, scripting can be reduced to a few prototype sentences [20]. The similarity between short texts is computed through the use of knowledge base such as the English WordNet. However due to the lack of resources in Urdu such as an appropriate WordNet, lexicons, annotated electronic dictionaries, corpora and well-developed ontologies that describe relationships among words and entities in written text [21] NLP and STSS are not appropriate methods to develop a Urdu CA. It should be noted that work has begun on the development of an Urdu WordNet [22], the work is still in very early stages and not developed enough to be deployed in a CA. the remaining technique PM is one of the most ubiquitous and popular methods for building systems that appear to be able to conduct coherent, intelligent dialogs with users [23]. The user utterance is matched to a database of pre-scripted patterns, rather than trying to understand the utterance. Once a pattern is matched a response is delivered back to the user. Creating scripts is a highly skilled craft and labour intensive task [1], requiring the anticipation of user utterances, generation of permutations of the utterances and generalization of patterns through the replacement of selected terms by wild cards. Modifications to rules containing the patterns can impact on the performance of other rules. The main disadvantage of pattern matching systems is the labour-intensive (and therefore costly) nature of their development. PM is a suitable method for developing an Urdu CA as it does not require extensive lexical resources to work.

Where have CAs been applied?

There is a variety of applications in which conversational agents can be used, one of the most widespread of which is information retrieval [24]. CAs have been deployed on websites, as helpdesk/customer service agents that respond to customers' inquiries about products and services [12]. Conversational agents associated with financial services' websites answer questions about account balances and provide portfolio information. Pedagogical conversational agents (also known as Intelligent Tutoring Systems) assist students by providing problem- solving advice as they learn [25] [26].

URDU LANGUAGE

There are fifty seven languages spoken in Pakistan. English is only understood by about 5% of this population. Therefore, for a Pakistani to benefit from the IT revolution (e.g. to give them access to services including e-government and e-commerce), solutions must be provided to this population in local languages [27]. Urdu is officially the national language of Pakistan, which houses about 180 million people. It is used in all official communication and government departments. Globally, Urdu is spoken by over 60 million people in more than 20. Urdu, an Indo-European language of the Indo Aryan family, is spoken in

India and Pakistan. Among all the languages in the world it is most closely similar to Hindi language. Urdu and Hindi both have originated from the dialect of Delhi region and other than minute details these languages share their morphology. Like Hindi has adopted many words from Sansikrit, Urdu has borrowed a large number of vocabulary items from Persian (Farsi) and Arabic [6]. Arabic and Farsi languages have close resemblance with Urdu, but Urdu is more complex as compare to Arabic and Farsi due to additional characters [28]. Urdu lies in the category of morphologically rich languages (MRLs) like Arabic, Persian, Chinese, Turkish, Finnish, and Korean. The MRLs pose considerable challenges for natural language processing, machine translation and speech processing [29].

The Challenges Faced in Developing a Urdu CA

10.3 Word order

One of the noteworthy aspects of Urdu grammar which has significant implications on the development of an Urdu CA is its word order. The basic word order of the Urdu Subject Object Verb (SOV) is an extremely common word order in the world's languages [30]. Although Urdu does conform to this rule it should be noted, that Butt [31] among others has highlighted that Urdu is non-configurational, that is, the ordering of elements of the sentence is not restricted. Bögel and Butt [32], provide further substance to this notion, they state that Urdu is a Free Word Order (FWO) language, meaning major constituents of a sentence can reorder freely [33] [34]. An example of this is illustrated in Figure 1 where all variations of the sentence are grammatically legitimate.

* Mujhe مجھے	neya نیا	shankthi card شناختی کارڈ	chahiye چاہئے ہیں
* Mujhe مجھے	shankthi card شناختی کارڈ	neya نیا	chahiye چاہئے ہیں
* Mujhe مجھے	shankthi card شناختی کارڈ	chahiye چاہئے ہیں	neya نیا
*Neya نیا	shankthi card شناختی کارڈ	chahiye چاہئے ہیں	mujhe مجھے
* Shankthi card شناختی کارڈ	neya نیا	chahiye چاہئے ہیں	mujhe مجھے
* Mujhe مجھے	chahiye چاہئے ہیں	neya نیا	shankthi card شناختی کارڈ

Figure 59 - Example of FWO (translation: I need a new ID card)

This varied word order is a significant issue in a pattern matching conversational agent. This is because the user utterance is pattern matched to a database of previously compiled responses. Pattern matching works by parsing a sequential string from beginning to end. In a language where there is no strict word order, it means that the domain will have to be scripted to compensate for all the different possible responses and variation in word order. This will result in extensive script writing which makes an already lengthy and time consuming task even more laborious.

10.4 Ambiguity

Like Arabic, Urdu vowels are indicated by marks (Diacritics) above and below the consonants [35]. In Urdu script, the consonantal context is clearly represented, but the vocalic sounds are represented (mostly) by marks or diacritics, which are optional and normally not written. Readers can guess the diacritics and thus can pronounce words correctly, based on their knowledge of the language. But un-diacritized Urdu text creates ambiguity for novice learners and computational systems [36]. An example of how diacritical marks inflect vocalic sounds on Urdu consonants in illustrated in Figure 2.

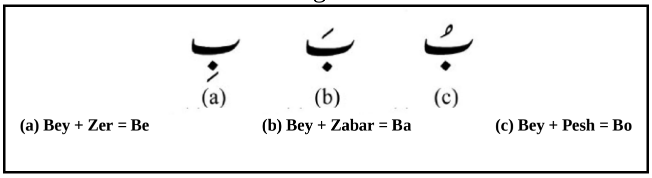


Figure 60 - Urdu Diacritical Marks

10.5 Morphology

Urdu style of writing does not have the concept of space to separate words. Similar to South-East Asian scripts like Lao, Thai and Khmer, Urdu readers are expected to segment the ligatures into words as they read along the text. In typing, space is used to get the right character shapes. Space is sometimes used within a word to break the word into constituent ligatures. However, if the ligature form is achieved without the use of space, it is sometimes not even used in between two words. Resulting in a visually correct sequence of two words for the readers but has no space between them. The notion of word spacing in Urdu is explained by Durrani [37] who states; the notion of space between words is completely alien in Urdu hand-writing. Children are never taught to leave space when starting a new word. They just tacitly use the rules and the human lexicon to know when to join and when to separate. This has implications on CA development and thus proper word segmentation must be done before strings are processed. Additionally, further challenges are posed due to the fact that there are no special rules syntax rules in Urdu, such as the use of capital letters in English, to indicate proper nouns names or the beginning of a sentence.

Knowledge Engineering the Domain

UMAIR was deployed a customer service representative for Pakistan's National Database and Registration Authority (NADRA) to answer customer queries on ID card applications and other related queries. The knowledge base for UMAIR was developed based on existing business logic used within this organisation. An interview was conducted an industry contact to gain some firsthand insight into the domain and the frequently arising issues they face. The interviewee was able to give firsthand insight into how queries are dealt with by their own customer service agents. The findings from the interviews were used to construct knowledge trees in order for them to be implemented in UMAIRs knowledge base. The knowledge base is made up of four layers: (1) domain specific contexts (2) Frequently asked questions (3) general chat (4) Urdu grammar data base. Layers 1-3 represent a state of the discussion UMAIR can be in; from this UMAIR is able to determine what the user wants from the discussion. Within each layer all the sub contexts related to that state are mapped together. The knowledge tree nodes are mapped to the contexts and all their related sub contexts through specialized conversational scripts. Operationally, UMAIR utilizes the scripts, along with the new PM engine to guide the user through the conversation to a predefined goal/leaf node, defined through the knowledge trees. Layer 4 contains Urdu grammar rules and words to help UMAIR classify and better understand the user utterance (e.g. questions, negative and positive statements, inappropriate words, valid words). UMAIR is able to utilize the knowledge base in order to deliver a coherent conversation to the user.

UMAIR

UMAIR is a PM, goal orientated CA which combines string similarity measures in order to converse in Urdu with the user to solve their queries related to the domain.

UMAIRs architecture consists of novel components which come together to handle the unique language specific difficulties in the Urdu language. Key features of the new architecture include the new PM engine which incorporates the WOW (Word Order Wizard) similarity algorithm and a Urdu scripting language. An overview of UMAIRs architecture is illustrated in Figure 3.

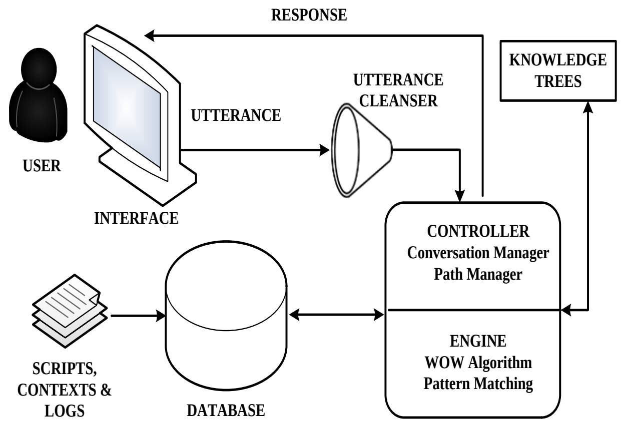


FIGURE 61 - UMAIR ARCHITECTURE

11.1 The Controller

The controller is responsible for directing and managing the entire conversation. The controller is the core of the CA and works with several other components to ensure the conversation goal is achieved. The controller is also responsible for delivering an intelligent, cohesive and goal led conversation.

The controller works together with the conversation and path manager to ensure the conversation is following the correct path, or switch context where necessary. The controller also checks the utterance for unacceptable and inappropriate words, if found it is able to warn the user accordingly. Once the utterance is processed the controller is responsible for delivering responses back to the user as well as any accompanying supporting material such as pictures or documents that may help the user and their query.

11.2 Conversation and Path Manger

The role of the Conversation Manager (CM) is to control the flow of the conversation. Depending on the context the CM loads a predefined path stored in the database that ensures the goal of each context within the domain is met during the conversation. The conversation manager ensures that the user stays on topic, and manages the switching of the contexts during the discussion by working together with the Path Manager (PM) component. The path manager loads a path that utilizes the decision trees within UMAIRs architecture and it directs the conversation toward the desired leaf node where the goal of the particular context is achieved. Another aspect handled by the PM is the ability to handle utterances that are not related to the current context of conversation. Goal-oriented CAs must employ mechanisms to manage unexpected utterances in a way that appears intelligent [38]. If the path manager receives an utterance that is not in the path of the current context, the path manager checks the user utterance with the FAQ knowledge layer then checks to see if the utterance matches other contexts within the database. Once a match is found the utterance is responded to, and then the user is brought back to the point where the conversation digressed and directed towards the goal again in order for the conversation to reach its conclusion.

11.3 Utterance Cleanser

The utterance cleanser is responsible for normalizing the user utterance by removing special characters from the user input such as diacritics (i.e. ◌ّ◌َ◌ِ◌ً) and punctuation (i.e. $, &, *, !, ?, “”, £). Moreover, the cleanser also ensures that the words are segmented correctly, by checking each individual word of the utterance with the Urdu grammar database. The cleansing ensures that only clean and consistent input is sent forward for pattern matching. This also makes scripting the domain easier as the scripter does not have to anticipate punctuation and or other diacritical marks which can be entered by the user.

11.4 Log File

UMAIR will utilize a long term memory/log file feature, which will allow it to store several variables and conversation related information in a database table. The information captured and stored in the database can be utilized to evaluate the system and track end user conversations.

11.5 Scripting Language

The foundations of UMAIR’s scripting language are based on the Info Chat scripting language. The scripting language includes a novel feature that allows it to provide supporting material to the user. Depending on the context and needs of the user the scripting language allows supporting material to be conveyed to the user in the form of images, application forms, maps etc. This adds another dimension of support and makes UMAIR seem more helpful and intelligent to the user, as opposed to just providing responses strictly in text form. This material is stored in the scripting database and once a rule is fired, if that rule has material to support the user’s query it is delivered to them through the interface. Another feature is the AllowYesNo rule in the scripting language. Certain questions can be answered with a simple yes or no answer within the system, however in some instances a yes/no answer is not sufficient enough for the system to be able to make a firm tree traversal decision. UMAIR is able to ask a linking question related to the context in order to extract further information. Figure 4 outlines an example of 1 of the patterns scripted.

Context General – Application Form
Rule – App_Form Pattern: * form do I need for new ID card Pattern: which form * for ID card Pattern: I need a form * ID card Pattern: * form for new ID card Response: The form to apply for an ID card is the POC form. You can either download a form, or visit your local NADRA office where you can pick one up. Switch Context: null Switch to: null Support material: poc_form.pdf Requires Vars: No Allow Yes/No

Figure 62 - Translated Example of Scripted Rule

11.6 WOW Algorithm

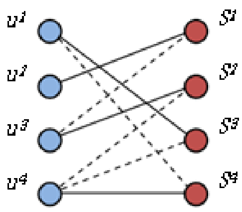
UMAIR introduces a novel method to determining the similarity between two sets of strings within CA's, while traditional CA's utilizes a PM based. UMAIR combines string similarity metrics and PM to overcome some of the intrinsic challenges in the Urdu language. Research found that one of the most prominent challenges that came with implementing the Urdu language in a CA was the issue of FWO. The biggest challenge of scripting CAs is the coverage of all possible user utterances [38]. This challenge grows considerably when a CA is implemented in the Urdu language as the FWO means one utterance can be said many different ways. The WOW algorithm is developed to tackle the issue of the FWO and reduce the need for scripting all possible word order variations of the same sentence. The WOW algorithm follows this procedure to calculate the similarity of the user utterance: **(1)** the user utterance and scripted pattern are split in to two separate token lists (*U* and *S*); **(2)** the first similarity check uses the Levenshtein edit-distance algorithm [39]. The edit distance is the total cost of transforming one string into another using a set of edit rules, each of which has an associated cost.

The calculation returns a score which is between 0 and 1. The closer the score is to 1 the higher the similarity. If the score gets a maximum value of 1 then the two tokens are identical. All the tokens in List *U* (utterance) and compared to the tokens in list *S* (scripted pattern). The highest matching score is then utilized as the edge weight (*E*) of that token. These token/node lists and edge weights make up a Bipartite Graph which is then utilized in the next step to compute the maximum similarity score. **(3)** The next step is to find a subset of node-disjoint edges that has the maximum total weight, the higher the total weight the closer the similarity of the two strings being compared.

A maximal weighted bipartite match is found for the bipartite graph constructed, using the Kuhn-Munkres Algorithm [40] – the intuition behind this being that every word in a sentence/utterance matches injectively to a unique word in the other sentence/pattern, if it does not then the highest match weight is utilized as that token/nodes edge weight (illustrated in Figure 4).

$$sim\ (u,p) = \frac{Maximum\ Sum\ of\ Bipartite\ Match}{Max\ (tokens\ (u), tokens(s))}$$

Eq. 1



w_{ij}	u^1	u^2	u^3	u^4
s^1	0	1	0.4	0
s^2	0	0	1	0.3
s^3	1	0	0	0.6
s^4	0.5	0	0	1

Figure 63 – Bipartite Graph and Edge Weight Matrix

The final similarity score (*sim*) between the sentences user utterance (*U*) and scripted pattern (*S*) is calculated through equation 1.

The WOW algorithm solves the complex word order issue that comes with the Urdu language by matching all possible word order variation on a single scripted pattern. Consequently it also significantly reduces the number of scripts that have to be scripted to deal with the issue of variation of word order in the Urdu language. It is duly noted that word order variation can change the meaning of the intended utterance, however to control such ambiguity features have been implemented to control the conversation through contexts. UMAIR is aware of the current context of the discussion, which helps overcome misunderstandings in word order as well as ambiguity through synonyms.

EXPERIMENTAL METHODOLOGY

Initial experiments have been conducted to evaluate the effectiveness and robustness of UMAIR and its components from an objective point of view. To formulate evaluation metrics, the Goal Question Metric (GQM) methodology was utilized [41]. The GQM methodology was implemented in order to highlight which metrics needed to be evaluated in order to gauge the effectiveness and robustness of UMAIR. A total of 24 participants were recruited all were residents of the Greater Manchester area, native Urdu speakers. The Participants were given scenarios that related to queries of ID card application. The participants spanned varying age groups and education levels and both genders were represented in the sample and all volunteered to participate for altruistic reasons. The participants were instructed to interact with UMAIR to resolve their particular query. The temporal memory/log file was then analyzed subsequent to the user's interaction. The log file provided backend insight into objective metrics related to the workings and success of the system and its associated algorithms.

RESULTS & DISCUSSION

Table 1 shows the results of the log file analysis.

Table 35 - Results of End User Evaluation

METRIC	UMAIR
Total number of utterances in all conversations	212
Average number of words per user utterance	5.0
Average number of utterances per conversation	8.8
Average conversation duration (mins)	3.2
Number of unrecognised utterances	12%
Percentage of conversations leading to acceptable goal	83.3%
Percentage of utterances containing word order variations of scripted patterns	33.6%
Percentage of conversations which reached goal without deviating the context	87%

The results demonstrated that the developed architecture and algorithms produced positive results. Table 1 reveals that 83% of conversations with UMAIR led to an acceptable goal. The conversations that didn't lead to a goal were mainly due to the users making spelling mistakes in their utterances, which meant the engine couldn't recognize them. Through the implementation of the novel WOW similarity algorithm UMAIR is able to deal with challenges of Urdu and PM all the word order variations on a single scripted pattern in the database, hence saving the scripter major time and effort. The results highlighted that 33% of all the user utterances contained valid word order variation of scripted patterns which were recognized and fired the appropriate rule associated with that script.

CONCLUSION & FUTURE WORK

The Urdu language posed many challenges when applied into development of an Urdu CA. This paper has outlined research to produce a new Urdu CA called UMAIR. It is the first Urdu CA, which contains novel features such as the WOW algorithm and scripting language in its architecture to deal with the language unique challenges of Urdu. The initial evaluation revealed positive results. Future work will concentrate on further enhancing the algorithms and knowledge base in order to strengthen UMAIRs conversation ability and utterance recognition. This will be followed by a within groups study with participants interacting with UMAIR and a human in a Wizard of Oz style experiment.

REFERENCES

[1] O'Shea, J., Z. Bandar, and K. Crockett, *Systems Engineering and Conversational Agents*, in *Intelligence-Based Systems Engineering*, A. Tolk and L. Jain, Editors. 2011, Springer Berlin Heidelberg. p. 201-232.

[2] Pickard, M.D., M.B. Burns, and K.C. Moffitt, *A theoretical justification for using embodied conversational agents to augment accounting-related interviews.* Journal of Information Systems, 2013.

[3] O'Shea, J., et al., *A comparative study of two short text semantic similarity measures*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2008, Springer. p. 172-181.

[4] Anwar, W., X. Wang, and X.-L. Wang. *A Survey of Automatic Urdu language processing*. in *Machine Learning and Cybernetics, 2006 International Conference on*. 2006. IEEE.

[5] Gordon, R.G., Jr., *Ethnologue: Languages of the World, Fifteenth edition*. 2005, SIL International. : Dallas, Tex.

[6] Hardie, A. *Developing a tagset for automated part-of-speech tagging in Urdu*. in *Corpus Linguistics 2003*. 2003.

[7] Sarfraz, H., et al., *Technology preparedness for disseminating flood relief and rehabilitation information to local stakeholders online: Lessons learnt while developing Punjab flood relief website in Urdu.* 2010.

[8] Sarfraz, H., A. Dilawari, and S. Hussain, *Assessing Urdu Language Support on the Multilingual Web.* 2011.

[9] Miller, G.A., *WordNet: a lexical database for English.* Communications of the ACM, 1995. **38**(11): p. 39-41.

[10] Hussain, S. and M. Afzal. *Urdu computing standards: Urdu zabta takhti (uzt) 1.01*. in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. 2001. IEEE.

[11] Ahmed, T. and M. Butt. *Discovering semantic classes for Urdu NV complex predicates*. in *Proceedings of the Ninth International Conference on Computational Semantics*. 2011. Association for Computational Linguistics.

[12] Rubin, V.L., Y. Chen, and L.M. Thorimbert, *Artificially intelligent conversational agents in libraries.* Library Hi Tech, 2010. **28**(4): p. 496-522.

[13] Massaro, D.W., et al., *Developing and evaluating conversational agents, Embodied conversational agents*. 2001, MIT Press, Cambridge, MA.

[14] Crockett, K., O.S. James, and Z. Bandar, *Goal orientated conversational agents: applications to benefit society*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2011, Springer. p. 16-25.

[15] Weizenbaum, J., *ELIZA—a computer program for the study of natural language communication between man and machine.* Communications of the ACM, 1966. **9**(1): p. 36-45.

[16] Marietto, M.d.G.B., et al., *Artificial Intelligence MArkup Language: A Brief Tutorial.* arXiv:1307.3091, 2013.

[17] Shawar, B.A. and E. Atwell, *A comparison between ALICE and Elizabeth chatbot systems.* 2002, Technical report, School of Computing, University of Leeds.

[18] Michie, D. and C. Sammut, *Infochat Scripter's Manual.* ConvAgent Ltd., Manchester, 2001.

[19] Chowdhury, G.G., *Natural language processing.* Annual review of information science and technology, 2003. **37**(1): p. 51-89.

[20] O'Shea, K., Z. Bandar, and K. Crockett. *A semantic-based conversational agent framework*. in *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*. 2009. IEEE.

[21] Naseem, T. and S. Hussain, *A novel approach for ranking spelling error corrections for Urdu.* Language Resources and Evaluation, 2007. **41**(2): p. 117-128.

[22] Zafar, A., et al. *Developing urdu wordnet using the merge approach*. in *Proceedings of the Conference on Language and Technology*. 2012.

[23] Bickmore, T. and T. Giorgino, *Health dialog systems for patients and consumers.* Journal of Biomedical Informatics, 2006. **39**(5): p. 556-571.

[24] Griol, D., J. Carbo, and J.M. Molina, *A statistical simulation technique to develop and evaluate conversational agents.* AI Communications, 2013. **26**(4): p. 355-371.

[25] Alobaidi, O.G., et al. *Abdullah: An Intelligent Arabic Conversational Tutoring System for Modern Islamic Education*. in *Proceedings of the World Congress on Engineering*. 2013.

[26] Latham, A., K. Crockett, and D. McLean, *An adaptation algorithm for an intelligent natural language tutoring system.* Computers & Education, 2014. **71**: p. 97-110.

[27] Hussain, S. *Computational Linguistics (CL) in Pakistan: Issues and Proposals*. in *EACL 2003*. 2003.

[28] Khan, K., et al., *An Efficient Method for Urdu Language Text Search in Image Based Urdu Text.* International Journal of Computer Science Issues(IJCSI), 2012. **9**(2).

[29] Abdul-Mageed, M. and M. Korayem. *Automatic identification of subjectivity in morphologically rich languages: the case of Arabic*. in *Proceedings of the 1st workshop on computational approaches to subjectivity and sentiment analysis (WASSA), Lisbon*. 2010.

[30] Whaley, L.J., *Introduction to typology: The unity and diversity of language*. 1997: Sage.

[31] Butt, M., *The structure of complex predicates in Urdu*. 1995: Center for the Study of Language (CSLI).

[32] Bögel, T. and M. Butt, *Possessive Clitics and Ezafe in Urdu.* Morphosyntactic Categories and the Expression of Possession, 2013. **199**: p. 291.

[33] Butt, M.J., T.H. King, and G.C. Ramchand, *Theoretical perspectives on word order in South Asian languages*. Vol. 50. 1994: Center for the Study of Language and Inf.

[34] Raza, G., *Subcategorization Acquisition and Classes of Predication in Urdu.* 2011.

[35] Alqrainy, S. and A. Ayesh, *Developing a tagset for automated POS tagging in Arabic.* WSEAS Transactions on Computers, 2006. **5**(11): p. 2787-2792.

[36] Raza, A. and S. Hussain. *Automatic diacritization for urdu*. in *Proceedings of the Conference on Language and Technology*. 2010.

[37] Durrani, N., *Typology of word and automatic word Segmentation in Urdu text corpus*. 2007, Citeseer.

[38] Latham, A.M., *Personalising Learning with Dynamic Prediction and Adaptation to Learning Styles in a Conversational Intelligent Tutoring System*. 2011, Manchester Metropolitan University.

[39] Ristad, E.S. and P.N. Yianilos, *Learning string-edit distance*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998. **20**(5): p. 522-532.

[40] Burkard, R.E. and E. Cela, *Linear assignment problems and extensions*. 1999: Springer.

[41] Fenton, N.E. and S.L. Pfleeger, *Software metrics: a rigorous and practical approach*. 1998: PWS Publishing Co.

Word Order Variation and String Similarity Algorithm to Reduce Pattern Scripting in Pattern Matching Conversational Agents

Mohammed Kaleem, James D. O'Shea MIEEE, Keeley A. Crockett SMIEEE

Department of Computing, Mathematics and Digital Technology
Manchester Metropolitan University
Manchester, United Kingdom
mohammed.kaleem@stu.mmu.ac.uk, (j.d.oshea, k.crockett)@mmu.ac.uk

Abstract— **This paper presents a novel sentence similarity algorithm designed to mitigate the issue of free word order in the Urdu language. Free word order in a language poses many challenges when implemented in a conversational agent, primarily due to the fact that it increases the amount of scripting time needed to script the domain knowledge. A language with free word order like Urdu means a single phrase/utterance can be expressed in many different ways using the same words and still be grammatically correct. This led to the research of a novel string similarity algorithm which was utilized in the development of an Urdu conversational agent. The algorithm was tested through a black box testing methodology which involved processing different variations of scripted patterns through the system to gauge the performance and accuracy of the algorithm with regards to recognizing word order variations of the related scripted patterns. Initial testing has highlighted that the algorithm is able to recognize legal word order variations and reduce the knowledge base scripting of conversational agents significantly. Thus saving great time and effort when scripting the knowledge base of a conversational agent.**

Keywords— Conversational Agents, Dialog Systems, Sentence Similarity, Urdu

INTRODUCTION

The term "Conversational Agent" (CA) is interpreted in different ways by different researchers; however the essence of CAs is natural language dialogue between the human and an application running on a computer [1]. Research into CA development has been focused on mainly English and western languages [2]. CA research and development into other languages such as Thai [3] and Arabic [2] is still in its early stages and languages such as Urdu do not have the extensive lexical infrastructures that are required to implement some CA components e.g. WordNet, and semantic measures [4]. Pattern Matching (PM) remains the predominant methodology for scripting the knowledge base that is utilized by the CA to converse with the user, as other development methodologies require sophisticated components which are still not readily available in other languages.

The traditional language for deployment of ICT solutions worldwide has been English, but it is evident that in order to reach the masses, the language medium needs to be one that is understood by the masses [5]. Urdu is a morphologically rich and a computationally resource poor language [6], consequently there are some challenges such as free word order to overcome in order to produce a functional Urdu CA. It is a well-known fact within the field of CA development that scripting is the most laborious and time consuming part of CA development [7, 8]. Moreover, script maintenance is another issue, as modifications to rules containing the patterns can impact on the performance of other rules. In a language such as Urdu the task of scripting and maintenance is further exacerbated due to the free word order of the language.

This paper outlines the novel WOW (Word Order Wizard) algorithm which was implemented in a new Urdu CA through which the challenge of scripting a free word order language in a CA is significantly reduced. The WOW algorithm processes the user utterances and the scripts at run time to calculate the similarity of the two sentences (utterance and scripted pattern) and check if the utterance is a valid word order variation of the scripted pattern.

This paper is structured as follows: Section II provides a brief overview of CAs, how they are developed and the challenges involved in their development. Section III outlines the Urdu language and the challenges it poses with relation to its implementation into a CA. Section IV provides a brief overview of the architecture of UMAIR the Urdu CA in which the WOW algorithm has been utilized. Section V is a detailed overview and walkthrough of the workings of the WOW algorithm. Sections VI and VII present the evaluation methodology, data collection results and evaluation results. Section VIII discusses the results, and finally Section IX presents the conclusions drawn from the research.

CONVERSATIONAL AGENTS

CAs essentially allow people to interact with computer systems intuitively using natural language dialogue [1]. In today's increasingly complex business environment, organisations face pressures regarding cost reduction, engagement scope, and attention to quality [9]. With this in mind, one of the most important emerging applications of CAs is online customer self-service/assistance, providing the user with the kind of services that would come from a knowledgeable or experienced human [7]. CAs of this nature are known as Goal Orientated-Conversational Agents (GO-CAs). GO-CA systems can provide anonymous, automated, interactive and consistent advice 24 hours a day in many different scenarios [10], for example helpdesk/customer service agents that respond to customers' inquiries about products and services [11]. Pedagogical conversational agents (also known as Intelligent Tutoring Systems) that assist students by providing problem- solving advice as they learn [2, 12].

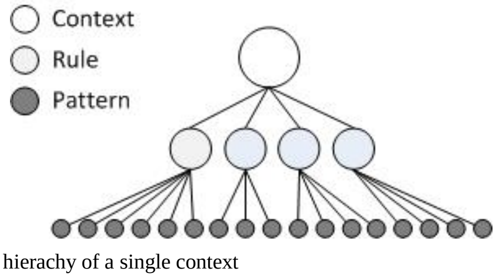
11.7CA Development

CAs have been developed using many different techniques. The three main techniques are Natural Language Processing (NLP) Short Text Semantic Similarity (STSS), and Pattern Matching (PM). NLP, STSS and PM are approaches that differ from Machine Translation (MT), as the aim of machine translation is to translate text in one language to another. Whereas, the CA approaches aim to process the text in order to understand it and formulate an appropriate response. NLP is an area of research that explores how computers can be used to understand and manipulate natural language text or speech to do useful things [13]. NLP assumes certain aspects for it to work effectively. The utterance is expected to be grammatically correct which usually it is not, incorrect sentences may be "repaired" but this adds computational overhead. Another point is that languages are very rich in form and structure, and contain ambiguities. A word can have more than one meaning (lexical ambiguity) or a sentence might have more than one structure (syntactic ambiguity/free word order), in light of this the NLP approach is not suitable to develop a CA in the Urdu language.

Another approach that is adopted in the development of CAs is the utilization of Short Text Semantic Similarity (STSS) measures to gauge the similarity between short sentences (10 – 25 words longs) [7]. Through employing sentence similarity measures, scripting can be reduced to a few prototype sentences [14]. The similarity between short texts is computed through the use of a knowledge base such as the English WordNet or text corpora. However due to the lack of resources in Urdu such as an appropriate WordNet, lexicons, annotated electronic dictionaries, corpora and well-developed ontologies that describe relationships among words and entities in written text [15] NLP and STSS are not appropriate methods to develop a Urdu CA. It should be noted that work has begun on the development of an Urdu WordNet [16], the work is still in very early stages and not developed enough to be deployed in a CA.

The remaining technique known as PM is one of the most popular methods for building systems that appear to be able to conduct coherent, intelligent dialogs with users [17]. The user utterance is matched to a database of pre-scripted patterns, rather than trying to understand the utterance. Once a pattern is matched an appropriate response is delivered back to the user.

PM CA's use a pre-complied repository of scripts, which are grouped into contexts (Illustrated in Fig. 1). Each context is made up of a number of rules. Each rule consists of a number of patterns and a linked response which make up the CA's knowledge base.



Scripting hierachy of a single context

Each rule is the sub-topic that relates to an attribute of the context that a user utterance may be matched with. Each rule can have a number of different patterns that are used to match it with a user utterance. Patterns consist of a collection of words and wildcard symbols (e.g. *), wildcards are used within patterns to match any number of words, broadening the rules to match utterances containing specific key phrases [18]. An example of a scripted rule is illustrated in Fig 2.

Context ID Card – Application Form
Rule – App_Form
Pattern: * form do I need to for a new ID card **Pattern**: * which form shall I fill * ID card **Pattern**: * need a form a new ID card **Pattern**: * form to apply for a replacement ID card
Response: To apply for a new ID card you need to fill a POC form.

Example of a single scripted rule

PM is a suitable method for developing an Urdu CA as it does not require extensive lexical resources, or grammatically correct or complete input to work. However, the major draw backs of the PM approach are the scripting process itself and the subsequent maintenance of the scripts.

Traditional CA scripting requires the script writer to consider every permutation of a user utterance that a user may send as input [8]. The PM approach requires precompiled scripts that define the conversation to be executed by a pattern-matching engine. Scripting is a time-consuming process, which takes no consideration of semantic content, it is focused solely on the structural form of the sentence. This requires the anticipation of all possible user utterances, generation of word order permutations of the utterances and generalization of patterns through the replacement of selected terms by wild cards. The main disadvantage of pattern matching systems is the labor-intensive (and therefore costly) nature of their development [1].

Furthermore, modifications to rules containing the patterns can impact on the performance of other rules. Consequently the entire database of scripts has to be reassessed in order to maintain the integrity of the scripted rules and avoid rule clashes and misfiring rules. This is a high maintenance and almost impossible process. In addition, different script writers possess differing levels of ability and as such this can prove to be an exasperating task [8]. An example of a PM CA is InfoChat. InfoChat implements a pattern matching approach using a sophisticated scripting language known as Pattern Script [19]. InfoChat scripting language is a rule-based language, using the type of rule structure shown in Fig.2 to handle the expected conversation.

A new PM CA for Urdu will have to address these challenges as well as challenges related to the language which are outlined in the following section.

THE CHALLENGES OF URDU

Urdu is the national language of Pakistan and a major language of India with more than 60 million first language speakers and more than 100 million total speakers in more than 20 countries. Urdu originated from various languages and is most strongly influenced by Arabic and Persian. Like both of these languages, Urdu is also written from right to left with a written script resembling Arabic [20]. Following several years of research and development activities, CAs in English, European and East Asian languages have become a popular area. However, South Asian Languages especially Urdu have received less attention [21].

The development of linguistic CA's has primarily been focused on English and other European Languages. There is limited existing research for the Urdu language and only one known Urdu CA is under development [22]. There have been many factors causing slow growth of Urdu software. One of the contributing factors has been the lack of standards for Urdu computing [23]. Ahmed and Butt [4] argue that one of the major bottlenecks for development is the lack of lexical resources available for the Urdu language, for example the Urdu language doesn't have the established electronic infrastructures that is taken for granted in English and other European languages, such as lexicons, annotated electronic dictionaries, corpora and well-developed ontologies that describe relationships among words and entities in written text [15].

One of the major challenges faced in developing an Urdu CA is the loose grammatical structure of the language. Butt [24] among others has argued that Urdu is non-configurational, that is, the ordering of elements of the sentence is not restricted [25]. Bögel and Butt [26], provide further substance to this notion, they state that Urdu is a free word order language, meaning major constituents of a sentence can reorder freely.

A single sentence in Urdu can be expressed in multiple ways and still be grammatically correct. Word order in Urdu is relatively free [27]. This notion is also shared by [28], who states Urdu is a free word order language. The verb in a sentence usually (but not always) comes last and its arguments are put in any order before it. An example of this is illustrated in Fig. 3 where the first variation is almost always used but the others are also legitimate.

* Mujhe مجھے	neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے
* Mujhe مجھے	shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے
* Mujhe مجھے	shankthi card شناختی کارڈ	chahiye چاہیے	neya نیا
*Neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے	mujhe مجھے
* Shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے	mujhe مجھے
* Mujhe مجھے	Chahiye چاہیے	neya نیا	shankthi card شناختی کارڈ

Valid word order variation in a single sentence

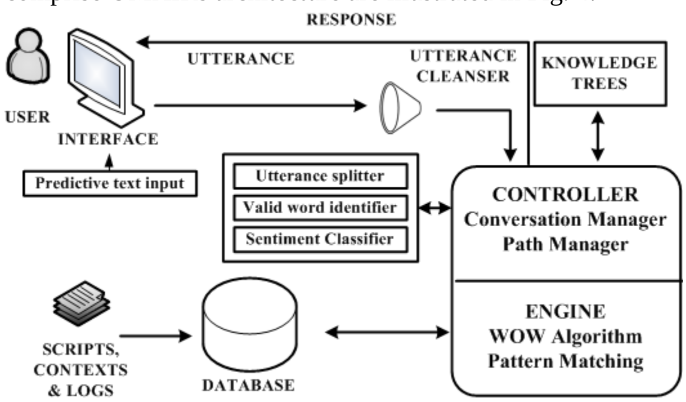
This type of word order variation is a significant issue in a pattern matching conversational agent. This is because the user utterance is matched to a database of previously compiled responses as discussed in the previous section. In a language where there is no strict word order, it means that the domain will have to be scripted to compensate for all the different possible responses and variation in word order. This means that the scripting could grow exponentially depending on the size of the selected domain. This will result in extensive script writing which make an already lengthy and time consuming task even lengthier and time consuming. The problem of scripting being a laborious task will be exacerbated when implementing a CA in Urdu.

As discussed earlier the biggest challenge of scripting CAs is the coverage of all possible user utterances [18]. This challenge increases if a CA is implemented in a language like Urdu as the free word order means one utterance can be said many different ways. This is a significant language specific issue; it would make scripting a CA in Urdu much more laborious task which would take significantly longer than scripting in a language with a fixed word order such as English.

It is evident that the word order rules in the Urdu language pose some novel challenges to overcome when implementing Urdu in a conversational agent. In light of the issues highlighted, a new methodology and algorithm is required to develop a novel conversational agent in the Urdu language, which can handle the language specific issues of this morphologically rich and resource poor language [29].

UMAIR ACHITECTURE

UMAIR is a PM, goal orientated CA, which includes string similarity measures in order to converse in Urdu with the user to solve their queries related to the domain ID card and passport application. UMAIRs architecture (illustrated in Fig. 4) consists of novel components which come together to handle the unique language specific difficulties in the Urdu language. Key features of the new architecture include the new PM engine which incorporates the WOW (Word Order Wizard) similarity algorithm and an Urdu scripting language. An overview of the components that comprise UMAIRs architecture are illustrated in Fig. 4.



UMAIR arcittecture overview

WOW ALGORITHM

UMAIR adopts a hybrid approach that combines string similarity metrics and traditional wild card PM, in order to overcome the inherent word order challenge in Urdu language. UMAIR's engine architecture comprises of components that work together to analyze the user utterance and provide the appropriate response. These components include a Wild Card PM Function combined with the WOW (Word Order Wizard) similarity algorithm which calculates similarity strength and handles the word order at run time. Similar approaches have been proposed in different contexts by [30, 31] however these approaches require vast lexical resources such as WordNets and lexical ontologies to compute the semantic similarity strength and to date, no reliable lexical knowledge base for Urdu exists [32]. The WOW algorithm was designed to be robust enough to handle changes in word order i.e. two strings which contain the same words, but in a different order, should be recognized as being similar. Furthermore significant sub string overlap should point to a degree of similarity, which compensates for common spelling variation in Urdu. Spelling variations are quite common in Urdu. The reason behind these variations is, there are many homophone characters (different letters representing the same phoneme) in Urdu (such as س and ص both represent a sound similar to S in English). People tend to confuse different homophones for each other, as a result, incorrect spelling of words having homophones becomes quite common [33].

The WOW algorithm similarity algorithm comprises of:

- Levenshtein Edit Distance Algorithm [34] used to calculate the similarity between two strings.
- Bipartite Matching [35] used to determine the word order variation.
- Kuhn-Munkres algorithm [36] (also known as the Hungarian method or the “matching problem”), used to find the maximum sum of a given matrix of weights.

The combination of these components within UMAIR’s engine come together to form a CA PM engine that calculates the similarity of the user utterance with scripted patterns using string similarity metrics in addition to taking word order into consideration. Therefore reducing the need to cover all possible word order variations when scripting the domain.

11.8WOW algorithm walkthrough

The WOW algorithm calculates similarity of the user utterance and scripted pattern in three steps by utilizing the algorithms described in the previous section. For this walk through assume the user utterance and database scripted pattern to be as follows:

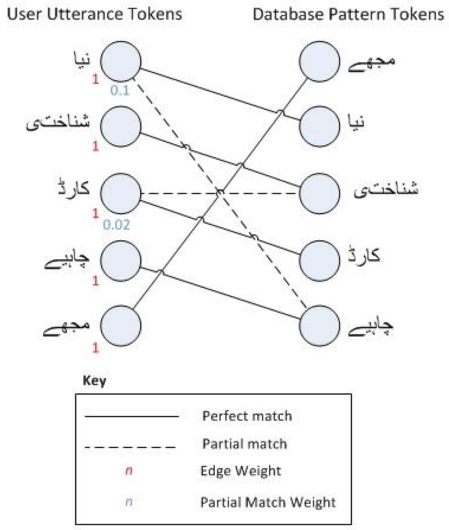
Utterance: مجھے نیا شناختی کارڈ چہیے
Pattern: نیا شناختی کارڈ چہیے مجھے

Both the user utterance and the database pattern translate to “I need a new ID card” however the utterance is in a different valid word order to the scripted pattern. This example is processed by the WOW algorithm as follows:

(1) Partition each string into a list of tokens after removing diacritical marks and punctuation, providing a bipartite graph. Tokens are separated firstly by whitespace characters and the each token is verified as a valid word through comparison to a database dictionary of Urdu words to ensure each word is split into valid Urdu word. As whitespace alone is not a reliable method for marking word boundaries in Urdu text [37].

user utterance: $u_1, u_2 \ldots u_n$
database pattern: $P_1, P_2 \ldots P_n$

(2) Given a graph G (U, P), G can be partitioned into two sets of disjoint nodes U (left tokens/utterance) and P (right tokens/pattern) such that every edge connects a node in U with a node in P, and each edge has a non-negative weight [38] which is determined by the edit distance. The weight of each edge which connects an *u1* to a *p1* is computed by the similarity of *u1* token and *p1* illustrated in Fig. 5.



Bipartie graph of utterance and scripted patten with edge wieghts

After the user utterance and pattern have been split in to two separate token lists, the first part of the similarity check uses the Levenshtein (*Lev*) edit-distance string matching algorithm [34]. The similarity method checks similarity the between the individual tokens in the two lists (i.e. user utterance and pattern from the database). The calculation returns a score which is between 0 and 1 for each token (illustrated in equation 1).

$$w(i, j) = Lev\ (token\ [u^n], token[p^n]) \quad \textbf{(1)}$$

The closer the score is to 1 the greater the similarity between the two tokens, which means that if the score gets a maximum value then the two tokens/words are identical. The maximum similarity score is then utilized as the edge weight. The results of this function are used to compute the weight (w) of edges which are then initialized and stored within a matrix of edge weights illustrated in Fig. 6.

$$\text{Database scripted pattern tokens}\ \left\{ \overbrace{\begin{pmatrix} w_{ij} & u_1 & u_2 & u_3 & u_4 & u_5 \\ p_1 & 0 & 0 & 0 & 0 & 1 \\ p_2 & 1 & 0 & 0 & 0 & 0 \\ p_3 & 0 & 1 & 0.2 & 0 & 0 \\ p_4 & 0 & 0 & 1 & 0 & 0 \\ p_5 & 0.1 & 0 & 0 & 1 & 0 \end{pmatrix}}^{\text{Utterance tokens}} \right.$$

Edge weight matrix

(3) The final task is to find a subset of node-disjoint edges that has the maximum total weight, the higher the total weight the closer the similarity of the two strings being compared. This is handled by the Kuhn-Munkres algorithm, the edge weights that are computed on step 2 are utilized to calculate the maximum sum of the edge weights. The final calculation returns the similarity strength between the two token lists which is a float value between 0 and 1. The closer the value is to 1 the stronger the similarity is between the two token lists. A value of 1 means the two token lists are identical, meaning all the words in the user utterance are present in the scripted database pattern in a different word order. A maximal weighted bipartite match is found for the bipartite graph constructed, using the Kuhn-Munkres Algorithm – the intuition behind this being that every keyword in a sentence/utterance matches uniquely to a unique keyword in the other sentence/pattern, if it does not then the highest match weight is utilized as that token/nodes edge weight.

Thus, the final similarity strength score (*sim*) between sentences user utterance (u) and pattern (p) is illustrated in equation 2.

$$sim\,(u,p) = \frac{Maximum\ Sum\ of\ Edge\ Weights}{Max\,(tokens\,(u), tokens(p))/2} \quad \textbf{(2)}$$

The algorithm takes in to account the number of words in the utterance and the pattern to ensure that all words are matched. If a word is missing from the utterance that is present in a pattern it reduces the final similarity score. If the score is below a set threshold level (.95) it is considered an unacceptable match. The threshold is set at .95 to compensate for common minor spelling variations found in the Urdu language.

Word order variation can change the meaning of the intended utterance, however to control such ambiguity the Urdu CA implements a conversation/path manager [22] to control the conversation through contexts. This helps overcome misunderstandings in word order as well as ambiguity through synonyms. The conversation/path manager allows the CA to be aware of the current context of the discussion through the scripting language which has variables stored within to let the conversation manager know which context the fired rule belongs to.

EXPERIMENTAL METHODOLOGY

The aim of the experiment was to test whether the WOW algorithm allowed the scripter to script a single pattern related to a single user utterance and have the algorithm detect all possible word order variations of that utterance and fire the corresponding rule as the response. A black-box [39] style experiment was conducted to gauge the robustness's and effectiveness of the WOW algorithm from an objective perspective. This was achieved by processing a number of utterances through the WOW algorithm and analyzing the output for accuracy and correctness. In order to gather data for the algorithm to process 10 user utterances/sentences/frequently asked questions were collated through interviews with a domain expert working for Pakistan's National Database and Registration Authority (NADRA) which processes all of the ID card and passport applications in the country. The sentences were printed on a sheet of paper and given to 40 participants as a survey with instructions to write all word order variations of each utterance/sentence they perceived to be legitimate word order variations of the original sentence. The responses from the participants were analyzed with an independent Urdu language expert who verified each legitimate word order variation. The verified sentences were run through the algorithm to evaluate the output. The sentences and the number of variations generated by the human participants are illustrated in Table 1.

RESULTS OF SURVEY

	Sentence	**Variations found**
1	مجھے نیا شناختی کارڈ چہیے I need a new ID card.	5
2	میں نے اپنا شناختی کارڈ کھو دیا ہے I have lost my ID card.	4
3	میرے پاس ان میں سے کوئی بھی دستاویزات نہیں ہے I do not have any of them documents.	4
4	مجھے کس فارم کو برنا ہو گا نیا شناختی کارڈ بنونے کے لیے؟ Which form should I fill in for a new ID card?	5
5	مجھے نیا پاسپورٹ چہیے I would like a new passport.	5
6	میں نے اپنا پاسپورٹ کھو دیا ہے I have lost my passport.	5
7	قریبی نادرا کا دفتر کہاں ہے؟ Where is the nearest ID card office?	4
8	ایک نئے شناختی کارڈ کتنے کا ہے؟ How much is a new ID card?	4
9	جہاں میں اپنی مکمل درخواست بھیجوں؟ Where do I send my completed application?	5
10	آج تم کیسے ہو؟ How are you today?	4
	Total	**45**

In total 45 different legitimate word variations were found from the 10 original sentences given to the participants. The variations of the sentences collated from the participants were then run through the WOW algorithm to test the accuracy of the algorithm i.e. whether or not the WOW algorithm correctly recognized them as word order variations of scripted patterns and fired the correct response rule.

RESULTS

The results of the black-box testing were captured in a log file. The results from the log file are summarized in Table 2.

RESULTS OF BLACK-BOX TESTING

Sentence	Expected number of times correct rule fired	Actual number of times correct rule fired
1	5	5
2	4	4
3	4	4
4	5	5
5	5	5
6	5	5
7	4	4
8	4	4
9	5	5
10	4	4

The results of the testing demonstrate that the WOW algorithm was able to recognize and correctly respond to all the 45 word order variations found from the 10 original sentences. In this case the scripting was reduced by 78% as only 10 patterns had to be scripted which covered 45 different word order variations which were not scripted but were correctly recognized and responded to by the WOW algorithm. Table 3 illustrates the results of a chi-square test of conducted to test whether there is a statistically significant relationship between the expected and actual outcomes of the results.

CHI-SQUARE TEST

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)
Pearson Chi-Square	10.000[a]	1	.002	
Continuity Correction[b]	6.400	1	.011	
Likelihood Ratio	13.863	1	.000	
Fisher's Exact Test				.008
Linear-by-Linear Association	9.000	1	.003	
N of Valid Cases	10			

a. 4 cells (100.0%) have expected count less than 5. The minimum expected count is 2.50.

b. Computed only for a 2x2 table

A chi-square test of independence of the relationship between the expected and actual outcomes of the testing finds a statistically significant relationship between the variables (expected and actual outcomes).

DISCUSSION

The WOW algorithm has allowed the Urdu CA to mitigate the complex word order issue that comes with the Urdu language. It also significantly reduces the number of patterns that have to be scripted to deal with the issue of word order an example of this is illustrated in Fig. 7. In Fig. 7 the first pattern is scripted in UMAIR and the remaining five patterns are not scripted, but are covered within the knowledge base with the WOW algorithm. Therefore, reducing the number of patterns that have to be scripted in the database, saving a significant amount of time, effort and furthermore makes the maintenance of scripts a much simpler endeavor.

As there are less patterns scripted in the database it reduces the chances of rule conflict which means maintenance is a lot less exasperating.

Scripted pattern	* Mujhe مجھے	neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے
Patterns covered	* Mujhe مجھے	shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے
	* Mujhe مجھے	shankthi card شناختی کارڈ	chahiye چاہیے	neya نیا
	*Neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے	mujhe مجھے
	* Shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے	mujhe مجھے
	* Mujhe مجھے	Chahiye چاہیے	neya نیا	shankthi card شناختی کارڈ

Scripted pattern and unscripted patterns covered by WOW

Fig. 7 illustrates how a single utterance can be said in many different ways in Urdu. This was a major challenge for the Urdu CA to overcome as this issue makes it very difficult for the scripter to script the domain, as all possible word order variations have to be pre-anticipated.

Subsequent to this evaluation the WOW algorithm was implemented in UMAIR in a real world application where it was tested through a larger scale end user evaluation which involved 24 participants. The results of the end user testing revealed that the WOW algorithm was found to reduce pattern scripting by 33% [22], meaning that 33% of the user utterances were unscripted word order variations of scripted patterns.

CONCLUSION

In a language with free word order such as Urdu the challenge of scripting the domain knowledge base is greatly amplified compared with a fixed word order language like English. The combination of the WOW algorithm and PM engine [22] implemented in UMAIR to process the user utterances has vastly reduced the need to script all possible word order variations of a single scripted pattern. The main objective behind the research and development of the WOW algorithm was to alleviate the complex word order issue that comes with the Urdu language by matching all possible word order variations on a single scripted pattern in order to reduce the time and effort required to script an Urdu conversational agent.

The novel WOW algorithm makes the job of the scripter easier, as all possible word order variations of scripted patterns do not have to be thought of and implemented. Only one pattern needs to be scripted and the rest are processed at run time by the algorithm.

The WOW similarity algorithm enables UMAIR to overcome the inherent challenges of developing a PM CA, and PM all the word order variations on a single scripted pattern in the database. Hence saving the scripter major time and effort. The algorithm can theoretically be applied to any language with free word order as it is based on PM principles, which means other CAs in languages with free word order such as Arabic, Hindi and Bangladeshi can utilize it.

REFERENCES

[1] O'Shea, J., Z. Bandar, and K. Crockett, *Systems Engineering and Conversational Agents*, in *Intelligence-Based Systems Engineering*, A. Tolk and L. Jain, Editors. 2011, Springer Berlin Heidelberg. p. 201-232.

[2] Alobaidi, O.G., K.A. Crockett, J.D. O'Shea, and T.M. Jarad. *Abdullah: An Intelligent Arabic Conversational Tutoring System for Modern Islamic Education*. in *Proceedings of the World Congress on Engineering*. 2013.

[3] Osathanunkul, K., J. O'Shea, Z. Bandar, and K. Crockett, *Semantic similarity measures for the development of thai dialog system*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2011, Springer. p. 544-552.

[4] Ahmed, T. and M. Butt. *Discovering semantic classes for Urdu NV complex predicates*. in *Proceedings of the Ninth International Conference on Computational Semantics*. 2011. Association for Computational Linguistics.

[5] Sarfraz, H., S. Hussain, M. Bano, and A. Dilawari, *Technology preparedness for disseminating flood relief and rehabilitation information to local stakeholders online: Lessons learnt while developing Punjab flood relief website in Urdu.* 2010.

[6] Syed, A.Z., M. Aslam, and A.M. Martinez-Enriquez, *Sentiment analysis of urdu language: handling phrase-level negation*, in *Advances in Artificial Intelligence*. 2011, Springer. p. 382-393.

[7] O'Shea, J., Z. Bandar, K. Crockett, and D. McLean, *A comparative study of two short text semantic similarity measures*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2008, Springer. p. 172-181.

[8] O'Shea, K., *Natural language scripting within conversational agent design.* Applied Intelligence, 2013: p. 1-9.
[9] Pickard, M.D., M.B. Burns, and K.C. Moffitt, *A theoretical justification for using embodied conversational agents to augment accounting-related interviews.* Journal of Information Systems, 2013.
[10] Crockett, K., O.S. James, and Z. Bandar, *Goal orientated conversational agents: applications to benefit society*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2011, Springer. p. 16-25.
[11] Rubin, V.L., Y. Chen, and L.M. Thorimbert, *Artificially intelligent conversational agents in libraries.* Library Hi Tech, 2010. **28**(4): p. 496-522.
[12] Latham, A., K. Crockett, and D. McLean, *An adaptation algorithm for an intelligent natural language tutoring system.* Computers & Education, 2014. **71**: p. 97-110.
[13] Chowdhury, G.G., *Natural language processing.* Annual review of information science and technology, 2003. **37**(1): p. 51-89.
[14] O'Shea, K., Z. Bandar, and K. Crockett. *A semantic-based conversational agent framework*. in *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*. 2009. IEEE.
[15] Naseem, T. and S. Hussain, *A novel approach for ranking spelling error corrections for Urdu.* Language Resources and Evaluation, 2007. **41**(2): p. 117-128.
[16] Zafar, A., A. Mahmood, F. Abdullah, S. Zahid, S. Hussain, and A. Mustafa. *Developing urdu wordnet using the merge approach*. in *Proceedings of the Conference on Language and Technology*. 2012.
[17] Bickmore, T. and T. Giorgino, *Health dialog systems for patients and consumers.* Journal of Biomedical Informatics, 2006. **39**(5): p. 556-571.
[18] Latham, A.M., *Personalising Learning with Dynamic Prediction and Adaptation to Learning Styles in a Conversational Intelligent Tutoring System*. 2011, Manchester Metropolitan University.
[19] Michie, D. and C. Sammut, *Infochat Scripter's Manual*. ConvAgent Ltd., Manchester, 2001.
[20] Abidi, A., I. Siddiqi, and K. Khurshid. *Towards searchable digital urdu libraries-a word spotting based retrieval approach*. in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. 2011. IEEE.
[21] Anwar, W., X. Wang, and X.-L. Wang. *A Survey of Automatic Urdu language processing*. in *Machine Learning and Cybernetics, 2006 International Conference on*. 2006. IEEE.
[22] Kaleem, M., K.A. Crockett, and J.D. O'Shea, *Development of UMAIR the Urdu Conversational Agent for Customer Service*, in *Proceedings of the World Congress on Engineering* 2014.
[23] Hussain, S. and M. Afzal. *Urdu computing standards: Urdu zabta takhti (uzt) 1.01*. in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. 2001. IEEE.
[24] Butt, M., *The structure of complex predicates in Urdu*. 1995: Center for the Study of Language (CSLI).
[25] Naim, C.M., *Introductory Urdu. rev*. Chicago: South Asia Language & Area Center, University of Chicago, 1999.
[26] Bögel, T. and M. Butt, *Possessive Clitics and Ezafe in Urdu.* Morphosyntactic Categories and the Expression of Possession, 2013. **199**: p. 291.
[27] Butt, M.J., T.H. King, and G.C. Ramchand, *Theoretical perspectives on word order in South Asian languages*. Vol. 50. 1994: Center for the Study of Language and Inf.
[28] Raza, G., *Subcategorization Acquisition and Classes of Predication in Urdu.* 2011.
[29] Mukund, S., D. Ghosh, and R.K. Srihari. *Using cross-lingual projections to generate semantic role labeled corpus for Urdu: a resource poor language*. in *Proceedings of the 23rd International Conference on Computational Linguistics*. 2010. Association for Computational Linguistics.
[30] Li, Y., D. McLean, Z.A. Bandar, J.D. O'shea, and K. Crockett, *Sentence similarity based on semantic nets and corpus statistics.* Knowledge and Data Engineering, IEEE Transactions on, 2006. **18**(8): p. 1138-1150.
[31] Bhagwani, S., S. Satapathy, and H. Karnick. *Semantic textual similarity using maximal weighted bipartite graph matching*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.
[32] Ahmed, T. and A. Hautli, *A first approach towards an Urdu WordNet.* Linguistics and Literature Review, 2011. **1**(1): p. 1-14.
[33] Ijaz, M. and S. Hussain. *Corpus based Urdu lexicon development*. in *the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan*. 2007.
[34] Ristad, E.S. and P.N. Yianilos, *Learning string-edit distance.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998. **20**(5): p. 522-532.
[35] Dasgupta, D., G. Hernandez, D. Garrett, P.K. Vejandla, A. Kaushal, R. Yerneni, and J. Simien. *A comparison of multiobjective evolutionary algorithms with informed initialization and kuhn-munkres algorithm for the sailor assignment problem*. in *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*. 2008. ACM.
[36] Burkard, R.E. and E. Cela, *Linear assignment problems and extensions.* 1999: Springer.
[37] Rashid, R. and S. Latif. *A Dictionary Based Urdu Word Segmentation Using Maximum Matching Algorithm for Space Omission Problem*. in *Asian Language Processing (IALP), 2012 International Conference on*. 2012. IEEE.
[38] Secer, A., A.C. Sonmez, and H. Aydin, *Ontology mapping using bipartite graph.* Int. J. Phys. Sci, 2011. **6**(17): p. 4224-4244.
[39] Myers, G.J., C. Sandler, and T. Badgett, *The art of software testing*. 2011: John Wiley & Sons.

UMAIR the Urdu Conversational Agent

M. Kaleem, J. O'Shea and K. Crockett
The Intelligent Systems Group, Department of Computing Mathematics & Digital Technology, Manchester Metropolitan University, Manchester, M1 5GD, UK.
Email: k4133m@gmail.com (j.d.oshea, k.crockett)@mmu.ac.uk

***Abstract*—This paper outlines the development of UMAIR an Urdu conversational agent developed as a customer service representative. UMAIRs architecture includes a novel engine, scripting language and WOW (Word Order Wizard) string similarity algorithm which are combined to tackle the language unique challenges of Urdu. Initial testing of the new architecture has yielded positive results towards UMAIR being able to cope with the inherent differences in the Urdu language such as word order.**

INTRODUCTION

The weakness in current Conversational Agent (CA) engines is that they are not suited to be implemented in other languages, languages with grammar rules and structure totally different to English. From a historical perspective conversational agents including the design of scripting engines, scripting methodologies, resources and implementation procedures have been implemented for the most part in English and other Western languages. Apart from the established work in English, initial research has taken place on Thai and Arabic CA development. But unfortunately South Asian Languages especially Urdu have received less attention, and to date there are no Urdu CA's. The research has found that the Urdu language does not have the computational resources that are readily available to western languages such as WordNets and lexical ontology's. This means the development of an Urdu Conversational Agent is not simply a matter of re-engineering existing methods with new content.

CONVERSATIONAL AGENTS

Conversational Agents (CAs) allow people to interact with computer systems intuitively using natural language dialogue [1]. One emerging application of CAs is online customer self-service/assistance, providing the user with the kind of services that would come from a knowledgeable or experienced human. Traditionally Conversational agents use a Pattern Matching (PM) technique to match user utterances to a repository of scripted pre-anticipated utterances and their appropriate responses. Over the years this method although reliable, has proven to be a laborious and time consuming task. Due to the grammatical nature of Urdu the laborious task of scripting becomes more challenging.

URDU LANGUAGE

Urdu is the national language of Pakistan and has more than 100 million total speakers in more than 20 countries [2]. One of the noteworthy aspects of Urdu grammar is that a single sentence in Urdu can be expressed in multiple ways and still be grammatically correct. Urdu is a free word order language [3].

UMAIR

UMAIR (Urdu Machine for Artificially Intelligent Recourse) is a PM, goal orientated CA which combines string similarity measures in order to converse in Urdu with the user to solve their queries related to the domain.

UMAIRs architecture consists of novel components which come together to handle the unique language specific difficulties in the Urdu language. Key features of the new architecture include the new PM engine which incorporates the WOW similarity algorithm and an Urdu scripting language. An overview of UMAIRs architecture is illustrated in Figure 1.

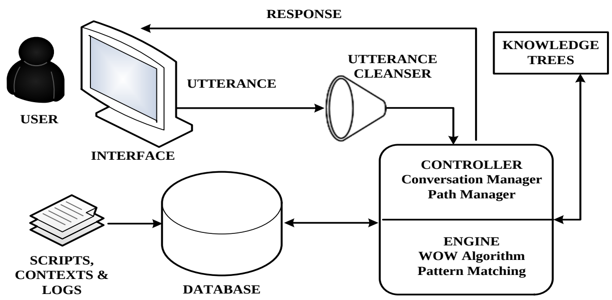


Figure 1 - UMAIR Architecture

The key contributions of the research include a novel engine, scripting language and WOW (Word Order Wizard) sentence similarity algorithms which are combined to tackle the language unique challenges of Urdu to produce, a customer service orientated CA. UMAIR was deployed in the selected domain of a passport and ID card advisor for NADRA (National Database and Registration Authority). Preliminary evaluation of the UMAIR has been conducted through Wizard of OZ testing and end user evaluation questionnaires. The evaluation and testing has yielded positive results, where 4 out of 6 evaluation metrics were not being perceived to be significantly different between the Wizard of OZ and UMAIR. Further testing of the WOW algorithm highlighted its ability to reduce the number of scripted patterns by up to 33%.

REFERENCES

1. O'Shea, J., Z. Bandar, and K. Crockett, *Systems Engineering and Conversational Agents*, in *Intelligence-Based Systems Engineering*, A. Tolk and L. Jain, Editors. 2011, Springer Berlin Heidelberg. p. 201-232.
2. Gordon, R.G., Jr., *Ethnologue: Languages of the World, Fifteenth edition*. 2005, SIL International. : Dallas, Tex.
3. Raza, G., *Subcategorization Acquisition and Classes of Predication in Urdu.* 2011.

Word Order Variation and String Similarity Algorithm to Reduce Pattern Scripting in Pattern Matching Conversational Agents

Mohammed Kaleem, James D. O'Shea MIEEE, Keeley A. Crockett SMIEEE
Department of Computing, Mathematics and Digital Technology
mohammed.kaleem@stu.mmu.ac.uk
Manchester Metropolitan University

Abstract

This research is concerned with a novel sentence similarity algorithm designed to mitigate the issue of free word order in the Urdu language. Free word order in a language poses many challenges when implemented in a conversational agent, primarily due to the fact that it increases the amount of scripting time needed to script the domain knowledge. A language with free word order like Urdu means a single phrase/utterance can be expressed in many different ways using the same words and still be grammatically correct. This led to the research of a novel string similarity algorithm which was utilized in the development of an Urdu conversational agent. The algorithm was tested through a black box testing methodology which involved processing different variations of scripted patterns through the system to gauge the performance and accuracy of the algorithm with regards to recognizing word order variations of the related scripted patterns. Initial testing has highlighted that the algorithm is able to recognize legal word order variations and reduce the knowledge base scripting of conversational agents significantly. Thus saving great time and effort when scripting the knowledge base of a conversational agent.

Introduction

The term "Conversational Agent" (CA) is interpreted in different ways by different researchers; however the essence of CAs is natural language dialogue between the human and an application running on a computer [1]. Research into CA development has been focused on mainly English and western languages [2]. CA research and development into other languages such as Thai [3] and Arabic [2] is still in its early stages and languages such as Urdu do not have the extensive lexical infrastructures that are required to implement some CA components e.g. WordNet, and semantic measures [4]. Pattern Matching (PM) remains the predominant methodology for scripting the knowledge base that is utilized by the CA to converse with the user, as other development methodologies require sophisticated components which are still not readily available in other languages.

Urdu is a morphologically rich and a computationally resource poor language [5], consequently there are some challenges such as free word order to overcome in order to produce a functional Urdu CA. It is a well-known fact within the field of CA development that scripting is the most laborious and time consuming part of CA development [6, 7]. o the free word order of the language. Moreover, script maintenance is another issue, as modifications to rules containing the patterns can impact on the performance of other rules. In a language such as Urdu the task of scripting and maintenance is further exacerbated due t

This poster outlines the novel WOW (Word Order Wizard) algorithm which was implemented in a new Urdu CA through which the challenge of scripting a free word order language in a CA is significantly reduced. The WOW algorithm processes the user utterances and the scripts at run time to calculate the similarity of the two sentences (utterance and scripted pattern) and check if the utterance is a valid word order variation of the scripted pattern.

Challenges of the Urdu Language

One of the major challenges faced in developing an Urdu CA is the loose grammatical structure of the language. Butt [9] among others has argued that Urdu is non-configurational, that is, the ordering of elements of the sentence is not restricted [10]. Bögel and Butt [11], provide further substance to this notion, they state that Urdu is a free word order language, meaning major constituents of a sentence can reorder freely. This is illustrated in Figure **1,** where all word order variations of the same sentence are valid.

* Mujhe مجھے	neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے
* Mujhe مجھے	shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے
* Mujhe مجھے	shankthi card شناختی کارڈ	chahiye چاہیے	neya نیا
*Neya نیا	shankthi card شناختی کارڈ	chahiye چاہیے	mujhe مجھے
* Shankthi card شناختی کارڈ	neya نیا	chahiye چاہیے	mujhe مجھے
* Mujhe مجھے	Chahiye چاہیے	neya نیا	shankthi card شناختی کارڈ

Figure 1: Valid word order variation in a single sentence

WOW Algorithm

The WOW algorithm calculates similarity of the user utterance and scripted pattern For this walk through assume the user utterance and database scripted pattern to be as follows:

Utterance: مجھے نیا شناختی کارڈ چاہیے
Pattern: نیا شناختی کارڈ چاہیے مجھے

Both the user utterance and the database pattern translate to "I need a new ID card" however the utterance is in a different valid word order to the scripted pattern. This example is processed by the WOW algorithm as follows:

(1) Partition each string into a list of tokens after removing diacritical marks and punctuation, providing a bipartite graph. Tokens are separated firstly by whitespace characters and the each token is verified as a valid word through comparison to a database dictionary of Urdu words to ensure each word is split into valid Urdu word. As whitespace alone is not a reliable method for marking word boundaries in Urdu text [12].

user utterance: $u_1, u_2 \ldots U_n$
database pattern: $P_1, P_2 \ldots P_n$

(2) Given a graph G (U, P), G can be partitioned into two sets of disjoint nodes U (left tokens/utterance) and P (right tokens/pattern) such that every edge connects a node in U with a node in P, and each edge has a non-negative weight [38] which is determined by the edit distance. The weight of each edge which connects an *u1* to a *p1* is computed by the similarity of *u1* token and *p1* illustrated in Figure 2.

(3) The results of this Step two are used to compute the weight (w) of edges which are then initialized and stored within a matrix of edge weights illustrated in Figure 3. The final task is to find a subset of node-disjoint edges that has the maximum total weight, the higher the total weight the closer the similarity of the two strings being compared. This is handled by the Kuhn-Munkres algorithm.

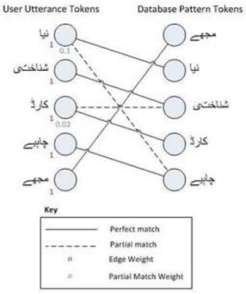


Figure 2

Utterance tokens

Database scripted pattern tokens

$$\begin{pmatrix} w_{ij} & u_1 & u_2 & u_3 & u_4 & u_5 \\ p_1 & 0 & 0 & 0 & 0 & 1 \\ p_2 & 1 & 0 & 0 & 0 & 0 \\ p_3 & 0 & 1 & 0.2 & 0 & 0 \\ p_4 & 0 & 0 & 1 & 0 & 0 \\ p_5 & 0.1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 3

Testing & Results

The aim of the experiment was to test whether the WOW algorithm allowed the scripter to script a single pattern related to a single user utterance and have the algorithm detect all possible word order variations of that utterance and fire the corresponding rule as the response. First a black-box [13] style experiment was conducted to gauge the robustness's and effectiveness of the WOW algorithm from an objective perspective. This was achieved by processing a number of utterances through the WOW algorithm and analyzing the output for accuracy and correctness.

The results of the black box testing demonstrate that the WOW algorithm was able to recognize and correctly respond to all the 45 word order variations found from the 10 original sentences. In this case the scripting was reduced by 78% as only 10 patterns had to be scripted which covered 45 different word order variations which were not scripted but were correctly recognized and responded to by the WOW algorithm.

Table 1 illustrates the results of a chi-square test conducted to test whether there is a statistically significant relationship between the expected and actual outcomes of the results. The chi-square test of independence of the relationship between the expected and actual outcomes of the testing finds a statistically significant relationship between the variables (expected and actual outcomes).

Chi-Square Test

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)
Pearson Chi-Square	10.000[a]	1	.002	
Continuity Correction[b]	6.400	1	.011	
Likelihood Ratio	13.863	1	.000	
Fisher's Exact Test				.008
Linear-by-Linear Association	9.000	1	.003	
N of Valid Cases	10			

Table 1: chi-square test of independence between expected and actual outcomes

Subsequent to the black box evaluation the WOW algorithm was implemented in UMAIR in a real world application where it was tested through a larger scale end user evaluation which involved 24 participants. The results of the end user testing revealed that the WOW algorithm was found to reduce pattern scripting by 33% [8], meaning that 33% of the user utterances were unscripted word order variations of scripted patterns.

Conclusion

In a language with free word order such as Urdu the challenge of scripting the domain knowledge base is greatly amplified compared with a fixed word order language like English. The combination of the WOW algorithm and PM engine [8] implemented in UMAIR to process the user utterances has vastly reduced the need to script all possible word order variations of a single scripted pattern. The main objective behind the research and development of the WOW algorithm was to alleviate the complex word order issue that comes with the Urdu language by matching all possible word order variations on a single scripted pattern in order to reduce the time and effort required to script an Urdu conversational agent.

The novel WOW algorithm makes the job of the scripter easier, as all possible word order variations of scripted patterns do not have to be thought of and implemented. Only one pattern needs to be scripted and the rest are processed at run time by the algorithm.

The WOW similarity algorithm enables UMAIR to overcome the inherent challenges of developing a PM CA, and PM all the word order variations on a single scripted pattern in the database. Hence saving the scripter major time and effort. The algorithm can theoretically be applied to any language with free word order as it is based on PM principles, which means other CAs in languages with free word order such as Arabic, Hindi and Bangladeshi can utilize it.

References

[1] O'Shea, J., Z. Bandar, and K. Crockett, *Systems Engineering and Conversational Agents*, in *Intelligence-Based Systems Engineering*, A. Tolk and L. Jain, Editors. 2011, Springer Berlin Heidelberg. p. 201-232.
[2] Alobaidi, O.G., K.A. Crockett, J.D. O'Shea, and T.M. Jarad. *Abdullah: An Intelligent Arabic Conversational Tutoring System for Modern Islamic Education*. in *Proceedings of the World Congress on Engineering*. 2013.
[3] Osathanunkul, K., J. O'Shea, Z. Bandar, and K. Crockett, *Semantic similarity measures for the development of thai dialog system*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2011, Springer. p. 544-552.
[4] Ahmed, T. and M. Butt. *Discovering semantic classes for Urdu NV complex predicates*. in *Proceedings of the Ninth International Conference on Computational Semantics*. 2011. Association for Computational Linguistics.
[5] Syed, A.Z., M. Aslam, and A.M. Martinez-Enriquez, *Sentiment analysis of urdu language: handling phrase-level negation*, in *Advances in Artificial Intelligence*. 2011, Springer. p. 382-393.
[6] O'Shea, J., Z. Bandar, K. Crockett, and D. McLean, *A comparative study of two short text semantic similarity measures*, in *Agent and Multi-Agent Systems: Technologies and Applications*. 2008, Springer. p. 172-181.
[7] O'Shea, K., *Natural language scripting within conversational agent design*. Applied Intelligence, 2013: p. 1-9.
[8] Kaleem, M., K.A. Crockett, and J.D. O'Shea, *Development of UMAIR the Urdu Conversational Agent for Customer Service*, in *Proceedings of the World Congress on Engineering* 2014.
[9] Butt, M., *The structure of complex predicates in Urdu*. 1995: Center for the Study of Language (CSLI).
[10] Naim, C.M., *Introductory Urdu. rev.* Chicago: South Asia Language & Area Center, University of Chicago, 1999.
[11] Bögel, T. and M. Butt, *Possessive Clitics and Ezafe in Urdu*. Morphosyntactic Categories and the Expression of Possession, 2013. **199**: p. 291.
[12] Rashid, R. and S. Latif. *A Dictionary Based Urdu Word Segmentation Using Maximum Matching Algorithm for Space Omission Problem*. in *Asian Language Processing (IALP), 2012 International Conference on*. 2012. IEEE.
[13] Myers, G.J., C. Sandler, and T. Badgett, *The art of software testing*. 2011: John Wiley & Sons.

2014 14th UK Workshop on Computational Intelligence (UKCI2014)

Manchester Metropolitan University
conv agent