Development of UMAIR the Urdu Conversational Agent for Customer Service

3 authors:

Mohammed Kaleem
Manchester Metropolitan University
4 PUBLICATIONS **9** CITATIONS

James Dominic O'Shea
Manchester Metropolitan University
78 PUBLICATIONS **1,195** CITATIONS

Keeley Crockett
Manchester Metropolitan University
163 PUBLICATIONS **1,797** CITATIONS

Proceedings of the World Congress on Engineering 2014 Vol I,
WCE 2014, July 2 - 4, 2014, London, U.K.

Development of UMAIR the Urdu Conversational Agent for Customer Service

M. Kaleem[1], J. O'Shea[1] and K. Crockett[1]

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**

I. 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.

[1] 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

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.

II. Conversational Agents

A. 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 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

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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 Chabot 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.

B. 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.

C. 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].

III. 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

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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].

IV. The Challenges Faced in Developing a Urdu CA

A. 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 چاہتے میں
* Mujhe مجھے	shankthi card شناختی کارڈ	neya نیا	chahiye چاہتے میں
* Mujhe مجھے	shankthi card شناختی کارڈ	chahiye چاہتے میں	neya نیا
*Neya نیا	shankthi card شناختی کارڈ	chahiye چاہتے میں	mujhe مجھے
* Shankthi card شناختی کارڈ	neya نیا	chahiye چاہتے میں	mujhe مجھے
* Mujhe مجھے	chahiye چاہتے میں	neya نیا	shankthi card شناختی کارڈ

Figure 1 - 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.

B. 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.

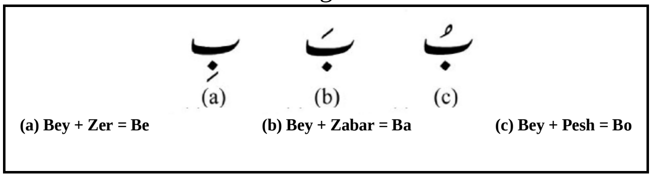


Figure 2 - Urdu Diacritical Marks

C. 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.

V. 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

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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.

VI. 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.

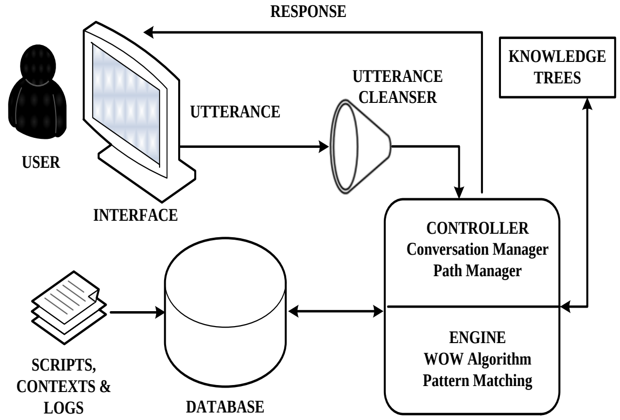


Figure 3 - UMAIR Architecture

VII. UMAIR ARCHITECTURE

A. 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.

B. 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.

C. 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.

D. 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.

E. 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.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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 4 - Translated Example of Scripted Rule

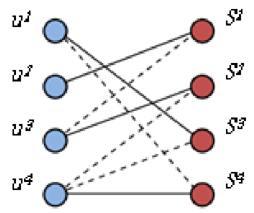
F. WOW Algorithm

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).

$$sim\,(u,p) = \frac{Maximum\ Sum\ of\ Bipartite\ Match}{Max\ (tokens\ (u), tokens(s))} \qquad \textbf{Eq. 1}$$



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 5 – 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.

VIII. 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.

IX. RESULTS & DISCUSSION

Table 1 shows the results of the log file analysis.

Table 1 - 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

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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.

X. 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.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)