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 tackle 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

I. 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 novel 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.

II. 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-CAs 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].

A. CA Development

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 [13]. 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 adds 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 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. 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 a 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.

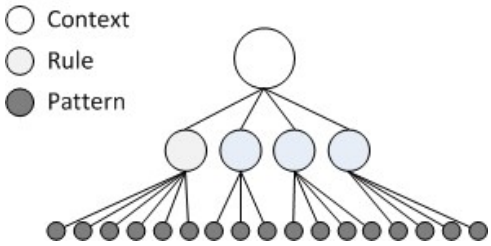

Fig. 1. Scripting hierachy of a single context

Each rule is the sub-topic that relates to a 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 general 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.

Fig. 2. 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, which depends on a rule structure 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.

III. 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 [20]. Urdu originated from various languages with most pronounced effects of Arabic and Persian. Like both of these languages, Urdu is also written from right to left with a written script resembling Arabic [21]. 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 [22].

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 [your paper] exists. 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. 1 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

Fig. 3. Valid word order variation in a single sentence

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 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 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. This means that the scripting could grow exponentially depending on the size of the selected domain.

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].

IV. UMAIR Achitecture

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 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.

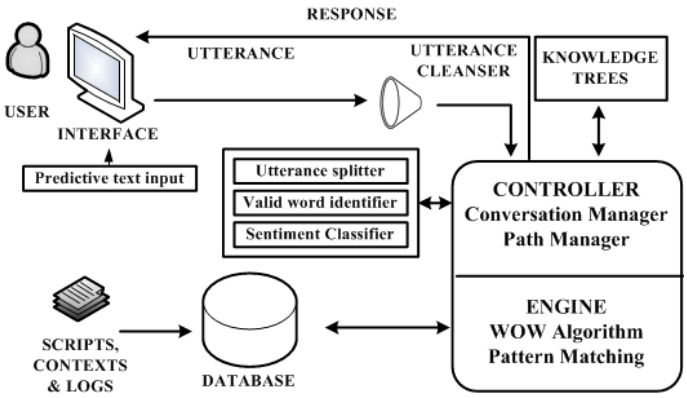

Fig. 4. Bipartie graph of utterance and scripted patten with edge wieghts

V. WOW ALGORITHM

UMAIR adopts a novel 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 WordNet's 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 of 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 variance.
- 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 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.

A. WOW 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 words are split into valid words. As whitespace alone is not a reliable method for marking word boundaries in Urdu text [37].

$$user\ utternce:\ u_1 u_2, \dots u_n$$
$$database\ pattern:\ p_1, p_2, \dots p_n$$

(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.

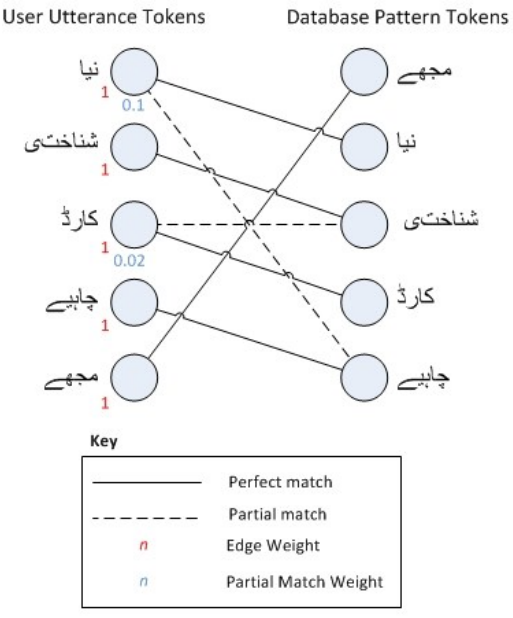

Fig. 5. Bipartie graph of utterance and scripted patten with edge wieghts

After the user utterance and pattern have been split in to two separate token lists, the first 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.

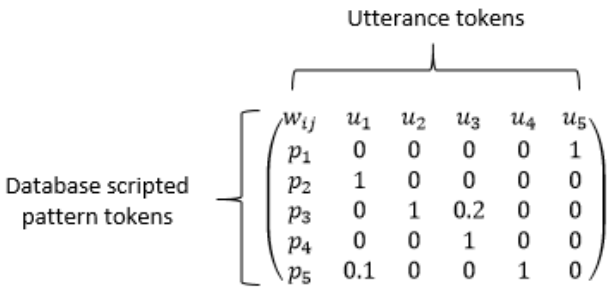

Fig. 6. Edge weight matrix

(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 by the Kuhn-Munkres algorithm that is used 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.

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

Word order variation can change the meaning of the intended utterance, however to control such ambiguity the Urdu CA implements a conversation/path manager [39] 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.

VI. 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 [40] 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 deals with 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.

TABLE I. Results of Survey

	Sentence	Variations found
1	مجھے نیا شناختی کارڈ چہیے	5
	I need a new ID card.	
2	میں نے اپنا شناختی کارڈ کھو دیا ہے	4
	I have lost my ID card.	
3	میرے پاس ان میں سے کوئی بھی دستاویزات نہیں ہے	4
	I do not have any of them documents.	
4	مجھے کس فارم کو برنا ہو گا نیا شناختی کارڈ بنونے کے لیے؟	5
	Which form should I fill in for a new ID card?	
5	مجھے نیا پاسپورٹ چہیے	5
	I would like a new passport.	
6	میں نے اپنا پاسپورٹ کھو دیا ہے	5
	I have lost my passport.	
7	قریبی نادرا کا دفتر کہاں ہے؟	4
	Where is the nearest ID card office?	

8	ایک نئے شناختی کارڈ کتنے کا ہے؟	4
	How much is a new ID card?	
9	جہاں میں اپنی مکمل درخواست بھیجوں؟	5
	Where do I send my completed application?	
10	آج تم کیسے ہو؟	4
	How are you today?	
	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.

VII. 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.

TABLE II. RESULTS OF BLACK-BOX TESTING

Sentence	Number of variations	Number of times correct rule fired	Accuracy
1	5	5	100%
2	4	4	100%
3	4	4	100%
4	5	5	100%
5	5	5	100%
6	5	5	100%
7	4	4	100%
8	4	4	100%
9	5	5	100%
10	4	4	100%

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.

VIII. Discussion

The WOW algorithm has allowed the Urdu CA to cope with 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. 4. In Fig. 4 the first pattern is scripted in UMAIR and the remaining five patterns are not scripted covered 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 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

Fig. 7. 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 its 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, it was found to reduce pattern scripting by 33% [39].

IX. Conclusion

In a language with free word order such as Urdu the challenge of scripting the domain knowledge base is greatly amplified than that of a fixed word order language like English. The combination of the WOW algorithm and PM engine [39] 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 solve 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.

X. 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., et al. *Abdullah: An Intelligent Arabic Conversational Tutoring System for Modern Islamic Education*. in *Proceedings of the World Congress on Engineering*. 2013.

[3] Osathanunkul, K., et al., *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., 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.

[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., 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.

[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., et al. *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] Gordon, R.G., Jr., *Ethnologue: Languages of the World, Fifteenth edition*. 2005, SIL International. : Dallas, Tex.

[21] 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.

[22] 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.

[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., et al., *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., et al. *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] 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.

[40] Myers, G.J., C. Sandler, and T. Badgett, *The art of software testing*. 2011: John Wiley & Sons.