Expert Systems With Applications 112 (2018) 342–352

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Intelligent conversation system using multiple classification ripple down rules and conversational context

David Herbert*, Byeong Ho Kang

Discipline of ICT, University of Tasmania, Hobart, Tasmania, Australia

ARTICLE INFO

Article history:
Received 27 November 2017
Revised 25 June 2018
Accepted 27 June 2018
Available online 28 June 2018

Keywords:
Knowledgebase systems
Textual question answering
MCRDR
Case based reasoning
Pattern matching

ABSTRACT

We introduce an extension to Multiple Classification Ripple Down Rules (MCRDR), called Contextual MCRDR (C-MCRDR). We apply C-MCRDR knowledge-base systems (KBS) to the Textual Question Answering (TQA) and Natural Language Interface to Databases (NLIDB) paradigms in restricted domains as a type of spoken dialog system (SDS) or conversational agent (CA). C-MCRDR implicitly maintains topical conversational context, and intra-dialog context is retained allowing explicit referencing in KB rule conditions and classifications. To facilitate NLIDB, post-inference C-MCRDR classifications can include generic query referencing – query specificity is achieved by the binding of pre-identified context. In contrast to other scripted, or syntactically complex systems, the KB of the live system can easily be maintained courtesy of the RDR knowledge engineering approach. For evaluation, we applied this system to a pedagogical domain that uses a production database for the generation of offline course-related documents. Our system complemented the domain by providing a spoken or textual question-answering alternative for undergraduates based on the same production database. The developed system incorporates a speech-enabled chatbot interface via Automatic Speech Recognition (ASR) and experimental results from a live, integrated feedback rating system showed significant user acceptance, indicating the approach is promising, feasible and further work is warranted. Evaluation of the prototype's viability found the system responded appropriately for 80.3% of participant requests in the tested domain, and it responded inappropriately for 19.7% of requests due to incorrect dialog classifications (4.4%) or out of scope requests (15.3%). Although the semantic range of the evaluated domain was relatively shallow, we conjecture that the developed system is readily adoptable as a CA NLIDB tool in other more semantically-rich domains and it shows promise in single or multi-domain environments.

1. Introduction

Conversational Agents (CA) and Natural Language Interfaces to Databases (NLIDB) systems typically require the system developer/author to have high-level skills in constructing either complex semantic or syntactic grammars, or highly technical scripting languages to parse user utterances, as well as database querying languages such as SQL. This introduces a clear, unwarranted separation between the system author and a domain expert – ideally *the domain expert* should be able to author and maintain the knowledge required by the system, but it is unreasonable to expect domain experts to have high-level technical or linguistic analysis skills (Androutsopoulos, Ritchie, & Thanisch, 1995; Smith, Crockett, Latham, & Buckingham, 2014). We propose a solution to this that allows an expert in the field to maintain knowledge that is used to create CAs with NLIDB capabilities. Our research uses a derivation of the knowledge engineering approach, Ripple Down Rules (RDR) (Compton & Jansen, 1990), called Contextual MCRDR (C-MCRDR).

RDR recognises the problem of eliciting knowledge from the domain expert – they have time constraints and they cannot usually provide a wholistic response in attempts to capture their knowledge (Biermann, 1998; Kang, Compton, & Preston, 1995). RDR removes this knowledge acquisition bottleneck by allowing the expert to build a knowledge-base (KB) incrementally as they only have to provide justification of a conclusion in a local context as it arises. We considered the application of RDR to the CA and NLIDB paradigms, and defined cases to be examples of user dialog – questions and statements that are relevant to the domain. The domain expert considers what should be appropriate responses – in RDR

* Corresponding author.
E-mail addresses: david.herbert@utas.edu.au (D. Herbert), byeong.kang@utas.edu.au (B.H. Kang).
URL: http://www.utas.edu.au/profiles/staff/ict/david-paul-herbert (D. Herbert), http://www.utas.edu.au/profiles/staff/ict/byeong-kang (B.H. Kang)

https://doi.org/10.1016/j.eswa.2018.06.049

terms, they are classifying each input case (which has pattern-matched components of the user utterance as attributes) by a response.

Standard RDR can only provide a single classification for each case, and a natural extension is to allow multiple classifications – Multiple Classification Ripple Down Rules (MCRDR) (Kang, 1995) is one such extension. In this paper we introduce C-MCRDR, which is a significant extension to MCRDR that facilitates constrained NL conversation via pattern-matching.

1.1. Contribution summary

The key features and contributions of C-MCRDR that facilitate CA and NLIDB services discussed in later sections are as follows:

1. Implicit retention of topical conversational context by adopting a stack-based modification to MCRDR's inference mechanism;
2. Intra-dialog contextual referencing via context-based variable definition and assignment (via regular expression pattern-matching) of relevant context that is maintained between dialog utterances;
3. Rule-count reduction and NLIDB via post-inference deferred classifications with database querying expressions (bound by relevant context variables);
4. Brittleness mitigation by:
   - (a) Pattern-matching of utterances to key terms using a lexical or phrasal paraphrasing approach;
   - (b) Utterance suggestion (rule lookahead) based on current topical context when an utterance is not recognised;
5. ASR transcription correction (when speech is used) by pre-processing terms using a set of corrective rules prior to inference;
6. Speech to Text (STT) correction by pre-processing terms using a set of corrective rules;
7. Dynamic rule maintenance of the live system courtesy of the RDR knowledge engineering approach

We conducted a usability evaluation study of a pilot system application of C-MCRDR and the results were very promising and positive, which is indicative the C-MCRDR approach to CAs and NLIDB is viable and worth further consideration. We will be further leveraging the system as a component in the command and control of autonomous systems via constrained NL.

The paper is organised by the following sections: Section 2 reviews related work associated around RDR and chat-based querying. We present C-MCRDR's modifications to standard MCRDR and the developed conversational system in Section 3. Sections 4, 5 and 6 detail the developed system's architecture, the methodology adopted in developing and evaluating the chat system, and the results of a pilot evaluation in a target domain respectively. We summarise the main results of this work together with proposals of future research in Sections 7 and 8.

2. Related work

2.1. Ripple down rules (RDR)

Ripple Down Rules (Compton & Jansen, 1990) arose from experiences the researchers had from maintaining a thyroid diagnosis expert system, GARVAN-ES1 (Horn, Compton, Lazarus, & Quinlan, 1985). During the maintenance of the original system they discovered that an approximate doubling of rule count in the KB only increased the accuracy of the system's diagnosis a couple of percentage points. It would typically take half a day for a new rule to be added due to the constraints of several factors, such as an expert endocrinologist's time, interpretation by the knowledge engineer, and extensive verification and validation to ensure the new rule did not compromise the existing KB. Instead, with RDR, the expert can add rules incrementally: they justify their new classification of a case in the context in which it arises. This is in contrast to other knowledge acquisition methods such as Repertory Grids (Gaines & Shaw, 1993), Formal Concept Analysis (Wille, 1992) and standard Case Base Reasoning (Aamodt & Plaza, 1994).

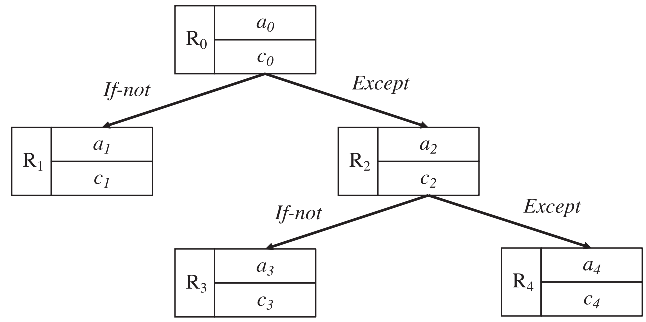

Fig. 1. RDR tree structure a_x – antecedent, c_x – consequent.

The original RDR structure is a binary tree, with each node comprised of a rule that consists of an antecedent and a consequent. During inference, the attributes of the current case to be classified are evaluated, starting at the root node (Fig. 1). If the antecedent's conditions are satisfied (a_0), evaluation passes to a child node (termed the *except* edge, here R_2). If the parent node's rule conditions are not satisfied, evaluation alternatively follows the other child edge, *if-not,* R_1. Either or both of the child nodes may not be present. Classification is the result of the last node to be satisfied. The root node (R_0) usually provides a default classification and a superficial antecedent to ensure all cases will be trivially assigned a class if no further rules are satisfied.

Multiple Classification Ripple Down Rules (MCRDR) (Kang, 1995) extends RDR's single classification inference outcome by allowing a case to have multiple classifications concurrently – in MCRDR's n-ary tree, inference considers all child nodes whose parent rules are satisfied, and evaluation concludes with each possible inference path terminated either by a satisfied leaf node or a satisfied node that has no satisfied children.

RDR and the MCRDR variants have had excellent research and commercial outcomes in the last two or more decades (Richards, 2009). For example, RDR and variants are used across diverse research and application areas: telehealth (Han et al., 2013); breast cancer detection (Miranda-Mena et al., 2006), legal text citation (Galgani, Compton, & Hoffmann, 2015); flight control systems (Shirazi & Sammut, 2008), robot vision systems (Pham & Sammut, 2005); induction (Gaines & Compton, 1992; 1995); clinical pathology reporting (Compton, 2011); and a help desk information retrieval mechanism (Ho Kang, Yoshida, Motoda, & Compton, 1997). For rapidity of development and implementation, (Han, Yoon, Kang, & Park, 2014) shows the MCRDR-backed KB methodology is closely aligned with the Agile software development approach.

2.2. Syntax and semantic parse trees

Very early systems focused on parsing an NL expression directly into *syntactic* parse trees, such as the often cited LISP-based LUNAR system (Woods, Kaplan, & Nash-Webber, 1972) where the parse tree maps to specific querying language expressions. Later hard-coded *semantic* grammars were used by systems such as LADDER (Hendrix, Sacerdoti, Sagalowicz, & Slocum, 1978), PLANES (Waltz, 1978) and CHAT-80 (Warren & Pereira, 1982) to analyse the input expressions to produce semantic concepts in the parse tree. These systems all suffered from poor inter-domain applicability; considerable effort is needed as grammars are complex and