User Guidance for Effective Query Interpretation in Natural Language Interfaces to Ontologies

Aliyu Isah Agaie, Masrah Azrifah Azmi Murad, Nurfadhlina Mohd Sharef, Aida Mustapha

Abstract—Natural Language Interfaces typically support a restricted language and also have scopes and limitations that naïve users are unaware of, resulting in errors when the users attempt to retrieve information from ontologies. To overcome this challenge, an auto-suggest feature is introduced into the querying process where users are guided through the querying process using interactive query construction system. Guiding users to formulate their queries, while providing them with an unconstrained (or almost unconstrained) way to query the ontology results in better interpretation of the query and ultimately lead to an effective search. The approach described in this paper is unobtrusive and subtly guides the users, so that they have a choice of either selecting from the suggestion list or typing in full. The user is not coerced into accepting system suggestions and can express himself using fragments or full sentences.

Keywords—Auto-suggest, expressiveness, habitability, natural language interface, query interpretation, user guidance.

I. INTRODUCTION

NATURAL LANGUAGE INTERFACES (NLIs) are designed to deal with natural language understanding of what the user wants and transform it into a computer language that specifies how to accomplish it. Each NLI system also has a scope and limitation which everyday users are unaware of. Most users have no knowledge of the structure of the ontology being queried and also lack the technical skills required in order to effectively deal with the structured information. It is therefore understandable that they may not see errors in their queries or even know how to write appropriate queries (according to the system's limitation and scope) in order to retrieve the correct information. This results in a mismatch between what the user expects of the NLI and the actual capabilities of the system [1], [2]. This mismatch (also referred to as habitability) is one of the major challenges faced by natural language interfaces to ontologies. According to [3], "habitability refers to how easily, naturally and effectively users can use language to express themselves within the constraints imposed by the system".

Queries are sometimes wrongly interpreted due to misspellings. More often than not, the query terms are expected to correctly match the ontology concepts and instance labels to be identified (the sequence in a text string must exactly match that of the backend, including whether the character is in upper or lower case). To illustrate this type of

Aliyu Isah Agaie, Masrah Azrifah Azmi Murad, Nurfadhlina Mohd Sharef and Aida Mustapha are with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia (e-mail: 1aaisah@live.com, masrah@upm.edu.my, nurfadhlina@upm.edu.my, aida_m@upm.edu.my).

problem, we shall use the "Juz Amma Structure Ontology" developed in [4]. If a user asks the question in Fig. 1, the system translates this into a machine understandable form and retrieves the answer. Fig. 2 (a) shows what transpires in the backend. However, if in writing the query, the user misspells "Medina" and instead writes "Madina", substituting "e" with "a", the system sees this as an error and therefore it cannot retrieve the answer (Fig. 2 (b)). The system does not recognize the term "Madina" and therefore underlines it in red color.
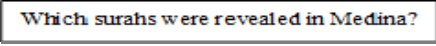
Which surahs were revealed in Medina?

Fig. 1 Example query

While some NLI systems attempt to automatically fix the errors in spellings, some other systems permit users to choose the ontology property names close to their intention from a list of suggestions. Both approaches have their setbacks. In the first, automatically fixing errors may lead to wrong interpretation where the supposed correction is also not right; whereas in the second, users may be confused with the property names used which result in them choosing from the suggestion list randomly and lead to inaccurate results. This paper is concerned about guiding the user to achieve accurate spellings in the first place.

A good interface is expected to support user expressiveness [5], [6]. However, NLIs typically support a more restricted language. To overcome this dilemma, in our previous study [7], we proposed that the best way to effectively retrieve information from ontologies is by guiding the users through the querying process while still providing them with an unconstrained (or almost unconstrained) way to query the ontology. The interface should support the users by guiding them to compose their queries through exploring the terms associated with the ontology being queried, presenting them with terminologies that are consistent with their mental understanding of the underlying ontology, and thereby enabling them to be able to make queries in order to obtain what they want.

By introducing an auto-suggest feature into the querying process, the user is assisted in formulating his query. Auto-suggest is quite helpful especially in complex information search activities where the user has no knowledge of the ontology being queried. It helps users to construct useful queries by exploring the domain terms and also aid in typing accuracy, thereby, increasing effectiveness of the search, and most certainly the overall efficiency too (by saving the time

that would be wasted in reformulating queries, if they are wrong). Guiding users to formulate their queries will result in better interpretation of the query, ultimately leading to an effective search [7]. The rest of this paper is organized as follows: Section II discusses on the related work. In Section III, we describe our approach. Section IV details the discussions and finally, Section V discusses on conclusions and future work.

(a)
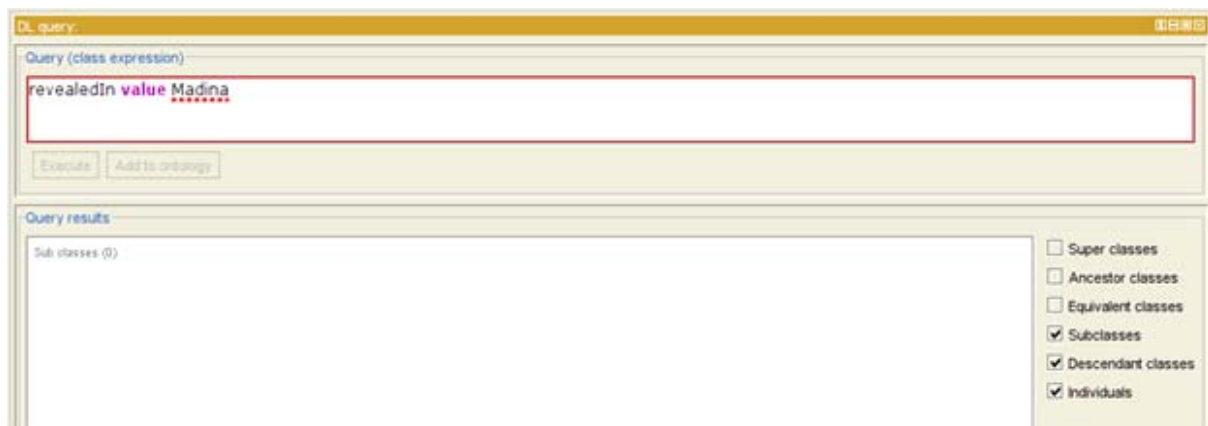
(b)

Fig. 2 (a) Query result for the question "Which surahs were revealed in Medina?" and (b) Failure due to spelling error, Medina is spelt as Madina

II. RELATED WORKS

The aim of guiding users to formulate their queries is to forestall the introduction of errors into the query during the query writing process. Several works have used different approaches in an attempt to achieve this objective. Reference [8] proposed the use of T9 spelling feature in combination with some color coding to aid dialogue with the system: blue color for user input, red for reporting errors, green for system output and orange color was used for clarification requests by the system. T9 is a good predictive text system but it is best suited for mobile devices such as tablets and phones. It also has the disadvantage of over generating words, which appear as "junk words" to users because of the optimized algorithm it uses that tries to achieve a compression ratio of 1byte for each word.

Revuelta-Martinez et al. presented some issues in [9] that needed to be addressed for an NLI to be seen as useful. They proposed the development of an environment that is interactive for the querying process. Their system made use of interactive text generation (ITG) to assist the user in formulating his query. The system employed the ITG technique in both the decoupled and partially coupled approaches. Both approaches are types of auto-complete systems, which help in transforming a problem of recall to that of recognition. Even though auto-complete helps in saving time and avoiding spelling errors, it is best suited for information retrieval tasks in a domain dependent system where the choices are limited: it provides assistance to carry out configured tasks. To adapt this approach to another domain will require heavy customization.

In order to guide a user in formulating his or her query, AskMe system [10] employed the use of auto-suggestion mechanism and lexical analysis to determine the correctness of the spelling of query terms. While the method is quite good, the problem is in the approach to building the system lexicon. The suggestions offered add a cognitive burden on

the user because they are mostly made up of special terms of entities and properties (labels) which are not consistent with the understanding of the users. As earlier stated, this may confuse the users and result in them choosing from the suggestion list randomly thereby leading to inaccurate results.

Ginseng [1] and Querix [11] were part of the systems used to evaluate the habitability hypothesis [2]. Ginseng [1] used an incremental parser that allowed users to complete words being typed by giving them suggestions to choose from in a pop-up menu and also predicted the next word. On the other hand, Querix [11] required users to use full sentences, which started with some limited sentence beginnings. Both approaches were too restrictive in that Ginseng did not accept terms that were not part of the suggestion list while Querix limited the users to questions starting with some certain beginnings. They both limit user expressiveness which is a key ingredient in developing user friendly NLIs [5], [6].

III. OUR APPROACH

This section describes the features that make up the suggestion mechanism, which is a sub system of the framework developed in [7]. For ease of understanding, the "Juz Amma Structure Ontology" [4] will be used as a case study.

A. Enhanced Concepts Store

This is the dictionary and is primarily made up of concepts/entities found in the ontology. The concepts/entities are extracted from the "the list of competency questions" document: this is the document used in eliciting the kind of queries a knowledgebase (KB) is expected to answer. These questions are derived from use-case scenarios and KBs are normally developed using the concepts/entities identified in the questions [4]. These terms are then enriched with terminologies that are consistent with the users' mental understanding.

It is noteworthy to state here that the "Juz Amma Structure Ontology" contains a lot of Arabic words written in English texts. This task is difficult, even for those that are conversant with the two languages due to the high variability in spellings when transcribing from Arabic to English. It has been identified that spelling variations of words in the same language may arise as a result of geographical distribution amongst speakers of that language and also due to variations in pronunciation [12]-[14]. Native Arabic speakers usually transcribe their words based on their spoken dialect while those who are not natives usually transcribe based on literal pronunciation [12], [13]. Therefore, in order to make the dictionary richer and more useful, almost all possible variations are added for the Arabic terms since people in various parts of the world do spell some particular words differently. Specifically, for the purpose of this research, we first took the transcribed concepts in the "Juz Amma Structure Ontology" and then added the transcriptions for the same concepts from [15], [16] (the spellings vary). For instance, in the ontology, the Arabic name for chapter 105 is transcribed as "AlFiil" while it is written as "Al Fil" and "Al-Fil" in [15] and [16] respectively. Thereafter, other variant terms that could represent these terms were added, as shown in Fig. 3. Thus, the dictionary is referred to as an Enhanced Concepts Store (ECS).

B. System Model and Query Construction Process

In this process, the user is guided to construct his query with respect to the words in the ECS. For example, in order to construct the query in Fig. 1, as the user types the query at the interface, the system automatically compares the words being typed with the content of the ECS and shows suggestions to the user. The user begins by typing "w". Since no word in the ECS starts with "w", the system allows the user to complete typing the word "which". A white space is then entered which signals the system to wait for another word. The word *sura* exists in the ECS, so as the user enters "s", the system pulls out all the words starting with "s", and then as the typing progresses, the words that best match the word being typed are continuously shown to the user in a wild card format. The user is able to choose from the list by clicking on the preferred word or select by highlighting with the cursor and pressing enter. The process is repeated for all the terms until the query construction is completed.

Fig. 4 shows the working of the auto-suggest feature presented in this paper. The system guides the user through the query construction process. Users are not forced to accept the system's suggestions, and are allowed to use terms that are not captured in the ECS.

AlFiil, Al Fil, Al-Fil, Al-Fiil, Fiil, Fil

Quran, Qur'an, Qoran, Qor'an, Kuran, Kur'an, Koran, Kor'an

Juz, Juz'i, Juzi, Juz'u, Juzu

Makkah, Makka, Maka, Macca, Maca, Mecca, Mekkah, Mekka, Meka
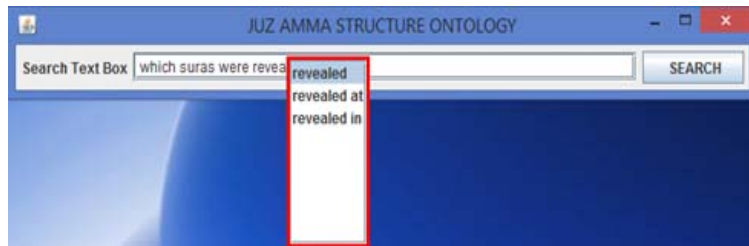
Medina, Medinah, Madina, Madinah

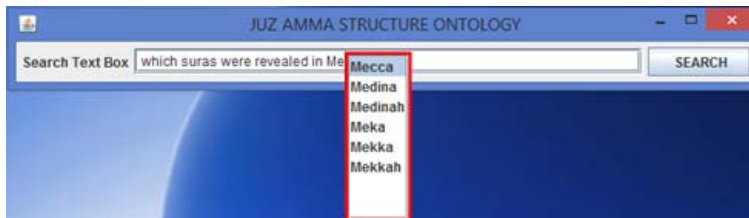Fig. 3 Examples of terms and their variant spellings
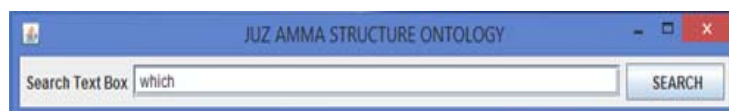
(a)
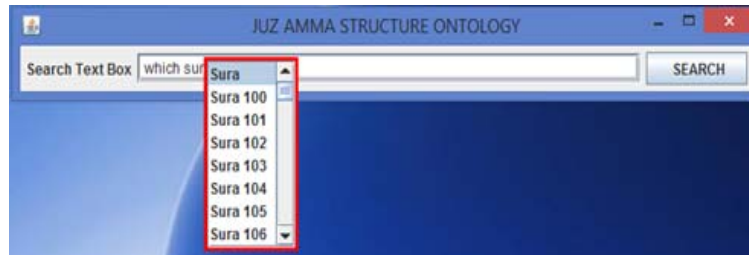
(b)

(c)

(d)

Fig. 4 (a)-(d) The system guides the user to type the question "Which suras were revealed in Medina?"

IV. DISCUSSION

Previous attempts by existing NLIs to guide users could be greatly improved if users are provided with an unconstrained (or almost unconstrained) way to query the ontology, allowing them to utilize the power of expression that is inherent in natural language. Guiding the users and at the same time giving them the freedom to express themselves is clearly a huge challenge. This is no mean feat but that is what our approach has achieved. The use of backend entity and property labels in [10] and the over generating of "junk words" in [8] add a cognitive burden on the user. To overcome this, our system uses the ECS, which is made richer and more useful by adding almost all possible spelling variations of the domain terms derived from "list of competency questions" document. These terms conform more to the users' mental understanding of the underlying ontology and thus, helps them to conceive and articulate more effective queries.

The coercing of users into accepting system suggestions in [1] and the limiting of users to only certain sentence beginnings in [11] completely hinder user expressiveness. Our approach is unobtrusive and subtly guides the users, permitting them to type in full or choose from the suggestion list. By forcing users to accept suggestions, [1] ensures that query terms correctly match the ontology concepts and instance labels: this is done at the expense of user expressiveness. Adding variant spellings of the terms to the ECS, employing WordNet [17] and using equivalent assertion to map query terms to backend labels (to be discussed in another work) will resolve the expressivity and cognitive burden issues.

When complex information search activities are involved, the effectiveness of the search can best be achieved by permitting the users to explore the KB terms while aiding them with the typing accuracy as in our approach. The ITG technique employed in [9] only provides assistance to carry out configured tasks. With [9], heavy customization is needed to adapt it to another domain.

V. CONCLUSIONS AND FUTURE WORK

Due to the scope and limitations of NLIs, they impose some restrictions on users. Our approach is an attempt at overcoming the constraints in order to ease users' information retrieval tasks. It allows users to construct useful queries while effectively expressing themselves using natural language. The users are guided to construct their queries by exploring the terms associated with the ontology being queried, presenting them with terminologies (captured in the

ECS) that are consistent with their mental understanding of the underlying ontology and thus, helps them to conceive and articulate more effective queries. It is unobtrusive and subtly guides the users, so that they have a choice of either selecting from the suggestion list or typing in full: the user is not forced into accepting system suggestions. The use of the ECS in our approach makes the portability of the system to any other ontology possible; only a little reconfiguration will be needed.

Our approach is yet to be tested in a real life scenario. We intend to evaluate the usability of the system with real users. Furthermore, since the suggestion mechanism is a sub-system of an overall design [7], subsequently, the next stage is to provide spelling correction and disambiguation mechanisms. This is because even when queries are correctly formulated, they may be misinterpreted due to spelling errors (as shown in Fig. 2) and language ambiguity. The proposed system will be designed to be capable of engaging the user in order to ascertain his intention by using robust disambiguation techniques.

REFERENCES

[1] A. Bernstein and E. Kaufmann, "GINO–a guided input natural language ontology editor," Semant. Web-ISWC 2006, vol. LNCS 4273, pp. 144–157, 2006.
[2] E. Kaufmann and A. Bernstein, "Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases," Web Semant. Sci. Serv. Agents World Wide Web, vol. 8, no. 4, pp. 377–393, 2010.
[3] D. Damljanović, M. Agatonović, H. Cunningham, and K. Bontcheva, "Improving habitability of natural language interfaces for querying ontologies with feedback and clarification dialogues," Web Semant. Sci. Serv. Agents World Wide Web, vol. 19, pp. 1–21, 2013.
[4] R. Iqbal, A. Mustapha, and Z. M. Yusoff, "An experience of developing Quran ontology with contextual information support," Multicultural Education & Technology Journal, vol. 7, no. 4, pp. 333–343, 2013.
[5] K. Elbedweihy, "Evaluating semantic search systems to identify future directions of research," Proceedings of the Second International Workshop on Evaluation of Semantic Technologies (IWEST), vol. 843, pp. 25–36, 2012.
[6] R. P. R and J. G. B, "Natural language interfaces to databases: An analysis of the state of the art," Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 463–480, 2013.
[7] A. I. Agaie, M. Azrifah, A. Murad, N. M. Sharef, and A. Mustapha, "A proposed framework for the development of an interactive natural language interface to ontologies," Journal of Convergence Information Technology (JCIT), vol. 9, no. 5, pp. 70–80, 2014.
[8] D. Calì, A. Condorelli, S. Papa, M. Rata, and L. Zagarella, "Improving intelligence through use of Natural Language Processing. A comparison between NLP interfaces and traditional visual GIS interfaces.," Procedia Computer Science, vol. 5, pp. 920–925, 2011.
[9] A. Revuelta-Martínez, L. Rodríguez, I. García-Varea, and F. Montero, "Multimodal interaction for information retrieval using natural language," Computer Standards Interfaces, vol. 35, no. 5, pp. 428–441, 2013.
[10] M. Llopis and A. Ferrández, "How to make a natural language interface to query databases accessible to everyone: An example," Computer Standards Interfaces, vol. 35, no. 5, pp. 470–481, 2013.
[11] E. Kaufmann, A. Bernstein, and R. Zumstein, "Querix: A natural language interface to query ontologies based on clarification dialogs," 5th ISWC, pp. 980–981, 2006.
[12] P. Dasigi and M. Diab, "CODACT: Towards identifying orthographic variants in dialectal arabic," Proc. 5th Int. Jt. Conf. Nat. Lang. Process, pp. 318–326, 2011.
[13] R. Al-Sabbagh and R. Girju, "YADAC: Yet Another Dialectal Arabic Corpus.," Lrec, pp. 2882–2889, 2012.
[14] C. M. Millward and M. Hayes, "A Biography of the English Language" Cengage Learning, 2011.
[15] A. Y. Ali, "The Meaning of the Glorious Quran: Text, Translation & Commentry," pp. 1–314, 2006.
[16] M. Pickthall and T. H. E. Opening, "Meanings of the Holy Qur'an," Word J. Int. Linguist. Assoc., pp. 1–209.
[17] "WordNet." (Online). Available: http://wordnet.princeton.edu/wordnet/ citing-wordnet/. (Accessed: 04-Oct-2015).