Handling Ambiguity Problems of Natural Language Interface for Question Answering

Omar Al-Harbi[1], Shaidah Jusoh[2], Norita Norwawi[3]

[1] **Faculty of Science and Technology, Islamic Science University of Malaysia, Malaysia**

[2] **Faculty of Science & Information Technology, Zarqa University, Zarqa, Jordan**

[3] **Faculty of Science and Technology, Islamic Science University of Malaysia, Malaysia**

Abstract

The Natural language question (NLQ) processing module is considered a fundamental component in the natural language interface of a Question Answering (QA) system, and its quality impacts the performance of the overall QA system. The most difficult problem in developing a QA system is so hard to find an exact answer to the NLQ. One of the most challenging problems in returning answers is how to resolve lexical semantic ambiguity in the NLQs. Lexical semantic ambiguity may occurs when a user's NLQ contains words that have more than one meaning. As a result, QA system performance can be negatively affected by these ambiguous words. In this research, we aim to resolve this problem by introducing CKCO (Context Knowledge & Concepts Ontology) approach. This approach integrates context knowledge and concepts ontology of a domain, into a shallow natural language processing (SNLP) technique. Concepts knowledge is modeled using ontology, while context knowledge contains a set of words with their senses obtained from WordNet Domain and a group of words within the proposed domain serve as context labels, and it is determined based on neighborhood words in the NLQ. We applied CKCO approach to a university QA domain for new students to examine the impact of WSD in retrieving correct answers. Experimental results show that the CKCO approach together with other components of our QA system yield a result which is 83.4% for precision. We focus on the ambiguity of nouns in the NLQ.

Keywords: *Question Answering (QA), Word Sense Disambiguation (WSD), Natural Language Processing (NLP), WordNet, Context Knowledge, Ontology.*

1. Introduction

Despite the great success of Web search engines, people still face the problem of how to retrieve what they really want. QA is a system that addresses this problem. The major aim of QA system in this research is to return a correct answer to a NLQ instead of a list of answers. QA is a task that combines techniques of information retrieval (IR), template matching, information extraction (IE), and natural language processing (NLP). A QA system is made up of 3 major modules: NLQ processing, documents processing, and answer processing [9], [10], and [11].

In QA, a NLQ is the primary source through which a search process is directed for answers. Therefore, an accurate analysis to the NLQ is required. One of the most difficult problems in developing a QA system is that it is so hard to find an answer to a NLQ [22]. The main reason is most QA systems ignore the semantic issue in the NLQ analysis [12], [2], [14], and [15]. To achieve a better performance, the semantic information contained in the NLQ should be considered in the NLQ analysis and answers processing. The NLQ processing is considered a fundamental component in the QA system, and its quality impacts the overall performance of the QA system.

Generally, QA can be categorized into two types based on the used methods. First, the shallow QA systems which use techniques like pattern matching in returning final answer. Such type of techniques ignores the issue of semantic, thus, many relevant answers could be missed or irrelevant answers could be retrieved [25], and [23]. Second, the deep QA systems which use techniques such as NLP. This type of QA considers the issues of semantic and syntactic analyses. According to [1] QA field has moved from only depending on retrieving and matching techniques to NLP techniques. However, NLP has been considered AI-complete problem because NLP requires extensive knowledge about the language and the ability to manipulate it [3]. The most challenging issue in NLP is a language is not free from an ambiguity problem.
Ambiguity means the capability of being understood in two or more possible senses. It is a pervasive phenomenon in human language [4]. Ambiguity has been recognized as a critical challenge in extracting semantic of a NLQ posed to a QA system [16]. In this research, we focus on lexical semantic ambiguity resolution in the NLQs posed to a QA system. Lexical ambiguity occurs when a word has more than one sense [3]. In QA, lexical ambiguity would cause confusion in interpretation of the

NLQ, and then affects negatively on the process of retrieving answers [2]. For example, given a NLQ "*How can a student deposit money into a bank?*", human knows that the *bank* here refers to a "financial institution". Whereas, given a NLQ "*How can a student join camping on the west bank?*", the *bank* here refers to the "sloping land beside the river". But unfortunately, it is very difficult for a computer to do the same job. Having more than one meaning for an individual word would lead to matching irrelevant answers and that will decrease the precision of retrieving the answers [2]. The typical solution to this issue is applying a Word Sense Disambiguation (WSD) technique to the NLQ analysis module. WSD refers to the process of deciding which of word's several senses is intended in a given context [3]. However, WSD itself is an open problem in the field of NLP [17]. Existing WSD methods either narrowly focus on a few specific words due to their reliance on expensive manually annotated training text, or give only mediocre performance in real-world settings [18].

This paper, proposes CKCO approach to overcome the lexical semantic ambiguity in the NLQs. To resolve the problem, we must consider the context and the domain knowledge in which each NLQ is posed. The proposed approach integrates context knowledge, and concepts knowledge of interesting domain, into a shallow natural language processing (SNLP) technique. Concepts knowledge is modeled using ontology, while context knowledge consists of words with their senses obtained from WordNet Domain and a group of words classified into predefined domains serve as context labels. Context knowledge is determined based on neighborhood words in the NLQ. SNLP includes implementing a chunker and shallow semantic analyzer.

2. Related Work

Work on QA is found in AI as computational linguistic, psychology, and philosophy [24]. QA systems have been evaluated and tracked in several academic workshops such as TREC [5], CLEF [6], and NTCIR [7]. Most of the research work conducted in QA utilize shallow approaches such as classification and matching words in NLQ with same words in retrieved texts. The need for effective techniques can precisely return correct answers made QA community to move towards many other new fields (e.g. NLP, knowledge representation (KR), and linguistic) [25]. A number of works that depend on NLP have been proposed in the past few years [16], [30], [31], and [32]. However, a few works have investigated the role of WSD in returning potential answers [8]. The automatic disambiguation of word senses is essential for a QA system [26]. Therefore, we survey some related work which investigated the role of WSD in QA systems.

The work [2] proposed a hybrid approach that combines WordNet with Internet as knowledge sources by which ambiguous words are disambiguated. This system ignored the context and the domain in which the question is posed, final decision of disambiguation starts with ranking possible senses based on search process through internet and number of search hits, and then any of two distinct words in a question are chosen. After that, they form every possible pair-combination of senses (synsets), one sense chosen from a word, the other sense chosen from the other word. Then they use all the synonyms of these two senses to form a query to search the Internet. Then, the pair of senses from two distinct words with higher number of hits is more likely to be the intended senses for each word. The accuracy of disambiguation is reported to be 35%.

The work [20] proposed an approach to determine the senses of words in queries by using WordNet. This approach is combined with information retrieval system to examine its effect in retrieving relevant documents. The approach makes use of WordNet by exploring synonyms, hyponyms, their definitions, and set of examples given to illustrate the use of the term in a particular sense. In this work, they gave less consideration to the context or domain issues in disambiguating words, e.g. the sense of *crime* word in a given query "*white collar crime sentence*" belongs to two domains, thus, they determined the correct sense '*an act punishable by law*' based on the domain of second sense of the word '*sentence*' which is accidentally match. This approach is sensitive to the exact wording of definitions, so the absence of a certain word can radically change the results. The accuracy of the method is 90%.

The work [21] introduced a natural language query engine that enables users to search for entities, relationships, and events that are extracted from biological literature. The work concerned mainly the step of NL query interpretation and translation. The syntactic and semantic query processing is guided by a domain ontology, which provides a mapping between linguistic structures and domain conceptual relations. In this work, linguistic ambiguity is resolved by identifying syntactic structure patterns which will be mapped to concepts to extract semantic relationships, thus, syntactic ambiguity might be faced. Additionally, the parser does not consider the role of context of NLQ in extracting semantic representation. The accuracy of this work is not reported.

The work [13] proposed a technique to resolve the ambiguous entities in entity extraction. This technique integrates subject and lexical knowledge, the possibility theory, and fuzzy sets into a statistical deep-NLP technique. In this work, context knowledge is generated by combining lexical meaning and input subjects that need to be entered along with a sentence by the user. For example, to resolve the ambiguous word *pen* the sense with maximum plausibility value will be attached, this value is assigned based on the entered subject. Suppose the user entered *livestock* as a subject, then sense *an enclosure in which babies may be left to play* will be assigned with the value 0.9, and other senses with lower value. In our work, beside we use knowledge about the domain, the user is not asked to enter the subject, as it will be automatically determined through unambiguous words in the NLQ.

In regard to WSD, the comparing between one system to another is complicated [26], and [27]. The reason is that every experiment has its own conditions and environment [26]. However, compared with the above QA systems, this research tries to resolve the lexical semantic ambiguity problem in a NLQ by considering the context in which the NLQ is posed, and make use of concepts that might be used within the proposed domain. This approach integrates contextual knowledge and concepts ontology into SNLP technique. SNLP include shallow syntactic analysis based on chunking mechanism, and sallow semantic analysis using semantic role labeling (SRL).

3. Proposed Approach

In this section, CKCO approach is introduced. The approach resolves lexical semantic ambiguity in NLQ by considering two pieces of knowledge: context knowledge, and concepts knowledge. The combination of these two pieces of knowledge is incorporated in a shallow NLP technique. Fig 1 illustrates the design of the approach, which can be divided into two main components namely: Shallow syntactic analysis, and shallow semantic analysis.

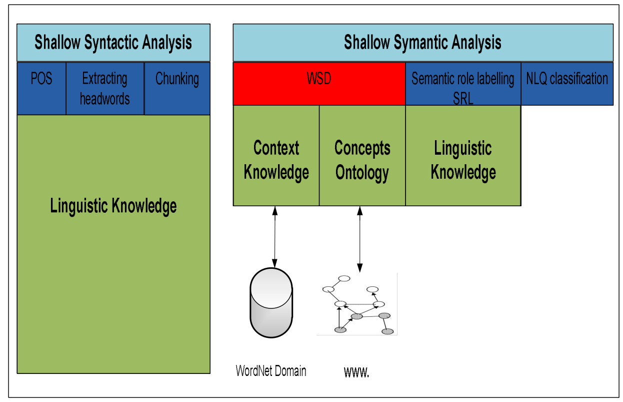

Fig. 1 The framework of a proposed approach.

3.1 Shallow Syntactic Analysis

The solution to lexical semantic ambiguity starts with shallow syntactic processing. This task consists of POS, extracting headwords, and chunking. The major goal of this component is to identify a syntactic structure of a NLQ can be attached with semantic information to obtain a formal semantic representation. For this task we apply a rule-based chunker.

3.1 Shallow Semantic Analysis

We apply a rule-based shallow semantic analyzer which is responsible for three major processes. First, word sense disambiguation, it is the core of this work. In this work, disambiguating lexical words mainly depends on context knowledge and concepts ontology. The semantic analyzer receives syntactic information from the chunker, and uses context knowledge to determine the relevant meaning based on context information that obtained from the posed NLQ to be matched with the stored context labels. After that, to identify the correct meaning among retrieved potential meanings, the analyzer uses knowledge about concepts within the proposed domain.

Second, once ambiguous words are resolved, this analyzer starts transforming the obtained syntactic and semantic information into semantic representation. In this research, semantic role labeling method (SRL) is applied to derive the final semantic representation of a NLQ. The semantic representation is represented by the predicate-argument structure, extended with some additional semantic information that assists in answer matching process. In this work, we consider one predicate per NLQ which is verbal predicates. The predicate-argument structure of our system is represented like $P\langle SRLs\rangle$ consists of a predicate P, and set of semantic role labels *SRLs*. A SRL is a binary structure $\langle w.sem, SRL \rangle$, consisting of the semantic meaning of a word *w.sem*. The meaning of an argument is assigned as follows: *Arg0* serves as a subject and *Arg1* is object. *Arg-Loc* represents a locational argument, *Arg-Time* represents the argumrnt of Time type, *Arg-Inst* represents instrumental argument, and *Arg-Manner* refers to a specific manner included in a sentence.

Third, NLQ semantic classification, this task is to classify the NLQ by its expected answer type (EAT). Knowing the answer type is important for finding the answer accurately.

3.1.1 Context Knowledge

The correct sense of a word in a NLQ relies on the context in which it is used. The context is determined based on the other words in the neighborhood in the NLQ. Thus, if the words *money*, *cash* or *teller* appears near the word *bank*, we can say that it is the *financial institute* and not the *sloping land*. This is called as local context. To use context in disambiguating words, a process of comparison between the local context and the domains of the target word's senses is performed. WordNet is not sufficient to perform such a comparison process. Moreover, WordNet is not a perfect resource for WSD, because it has the problem of the finedgrainedness of WordNet´s sense distinctions [26].

Therefore, we model a context knowledge resource to disambiguate lexical words. Context knowledge contains a set of words labeled with their senses and domain labels. The set of words with their senses and domain labels are manually obtained from the WordNet Domains lexical resource. WordNet Domains [33] is an attempt to extend the coverage of domain labels within an already existing lexical database which is WordNet. As a result WordNet Domains can be considered an extension of WordNet in which synsets have been annotated with one or more domain labels, selected from a hierarchically organized set of about 200 labels. In this work, WordNet Domain is used to decide if a lexical entry is ambiguous or not, determine synonyms, and to provide the context knowledge with the set of possible senses and domain labels of an ambiguous word.

For example, the word *bank* has 10 meanings; as illustrated in Table 1, each sense is labeled with a context that refer to a specific domain that include a set of relevant words. As we notice from the Table 2 one context label can be assigned to multiple senses for a word. Thus, to determine the correct meaning, knowledge about lexical meanings and its context are mapped to concepts ontology, which is described in the next subsection.

Table 1: Context knowledge of the word *bank*

Sense	*sense*	*Context*
#1	financial institute	Economy
#2	sloping land	Geography, Geology
#3	container	Economy
#4	the funds held by gambling house	Economy, Play
#5	a flight maneuver	Transport
#6	a supply or stock held in reserve	Economy
#7	a long ridge or pile	Geography, Geology
#8	a building in which the business of banking transacted	Architecture, Economy
#9	Bank building	Architecture, Economy
#10	bank, cant, camber (a slope in the turn of a road. . .)	Architecture

3.1.2 Concepts Ontology

Concepts knowledge is ontology consists of a set of concepts and relationships within the proposed domain. Ontology is becoming the essential methodology to represent domain-specific conceptual knowledge in order to promote the semantic capability of a QA system. In this work, the ontology is a framework that represents concepts and the relationships that exist among those concepts within the proposed domain. Furthermore, this ontology describes how concepts are related to linguistic knowledge such as lexicons. Concepts represent entities within the proposed domain such as *student*, *college*, *money*, *bank*, etc. Relationships represent the interactions among the concepts. Linguistic knowledge represents how senses that are extracted from WrodNet Domain associated with existing concepts.

Fig 2 illustrates a part of ontology of a university domain. The ontology is represented as a graph that consists of nodes (concepts) and edges (relationships), and the dashed arrow illustrate a relationship among three concepts.

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012
ISSN (Online): 1694-0814
www.IJCSI.org

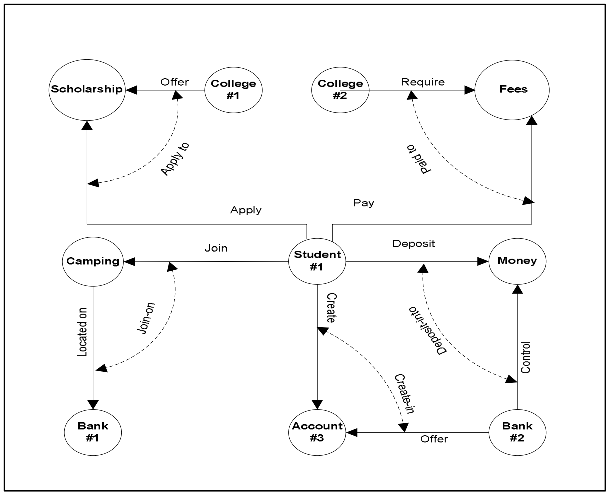

Fig. 2 An example of concepts-domain ontology.

4. Architecture of the QA System

The CKCO approach is packed into a QA system prototype. Fig 3 illustrates the fundamental architecture of the QA system, and our proposed approach is shown in the natural language interface, more specifically, in the NLQ processing phase. A user poses a NLQ through the interface going into a sequence of processes. There are 4 major modules that illustrate the architecture of the overall QA system, including CKCO approach. As mentioned earlier, this research concerns only with the user interface and answer processing modules.

4.1 User Interface

The user interface is a natural language interface, which the user's NLQ can be entered into and the answers returned back to the user. The user may pose the NLQ in different ways, e.g. yes/no questions, imperative questions, or wh-questions. In this work, only wh-questions are considered. This system can automatically answer English NLQs that are asked by new students about a university domain. In this work, we focus on NLQs containing multiple words, if the NLQ consists of a single word and the word has multiple senses, it is usually impossible to determine the correct sense of the word. A word that is classified as *noun* is defined as an entity and that is classified as a *verb* is defined as a relation. A NLQ is made up of a sequence of entities. These entities can be ambiguous or unambiguous. Although, this research focuses on ambiguous ones, the unambiguous entities will serve in determining the local context of ambiguous entities. In the user interface, there are some tasks are carried out; the tasks are: chunking, semantic processing, and answer type recognition. The following subsections explain these tasks.

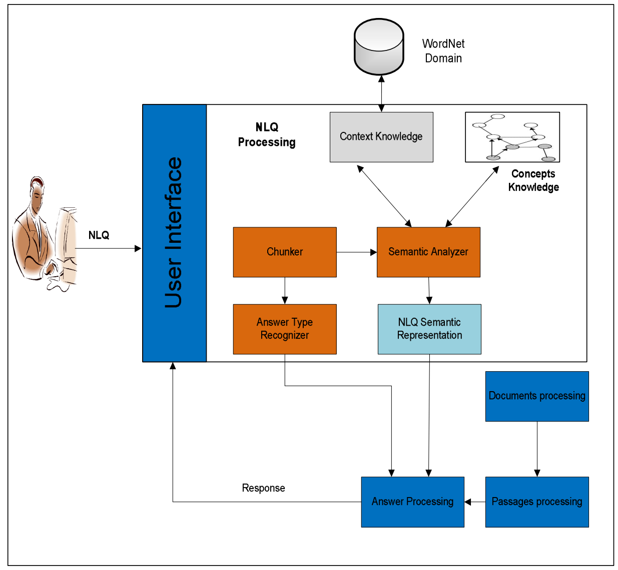

Fig. 3 The architecture of QA system.

4.1.1 Chunker

In this module, the NLQ processor firstly performs the step of POS. The proposed rule-based tagger reads the NLQ and assigns a class to each word, such as *noun*, *verb*, *adjective*, *etc*. For this task, we provide the tagger with the necessary linguistic knowledge. For example, given a question "*How can student deposit money into the bank?*", can be tagged as follows:

[How/Wh-Q] [can/Aux] [student/Noun] [deposit/Verb] [money/Noun] [into/ IN] [the/Det] [bank/Noun]

In this step, each word which is classified as *verb* or *noun* is extracted as a headword. Words with *noun* category are identified as an entities *E*, and a word with *verb* category as a relation *R* between two or more entities. For example, the NLQ "*How can a student deposit money into the bank?*" contains *student*, *money*, and *bank* words as entities; the word *deposit* will be defined as a relation between entities. A NLQ is made up of a sequence of entities.

After having POS for each word, the processor groups the word as constituents (e.g. Noun Phrase (NP), Verb Phrase (VP), and Prepositional Phrase (PP)). For this task, we built a rule-based chunker, which receives a sequence of tagged words, and then divides the NLQ into syntactically correlated segments. For example, given a question "*How can student deposit money into the bank?*", can be divided as follows:

[student/NP] [deposit/VP] [money/NP] [into the bank/PP]

4.1.2 Semantic Analyzer

Semantic analyzer receives the syntactic information from the chunker. Syntactic information includes entities and relationships, and chunks. Nouns are examined by the context knowledge to detect the ambiguity of lexical. Thus, *bank* is detected as an ambiguous word, whereas other unambiguous words in the NLQ are used to indicate to the local context of the ambiguous word. For example, the word *money* is identified as an unambiguous word, and the word *money* is classified into Economy domain, thus, the processor looks up the context knowledge to find *bank's* senses labeled with Economy context. According to Table 1, there are four senses (#2, #3, #4, #6, and #8) labeled with Economy context have been considered.

In order to filter out the retrieved senses and decide the correct sense, concepts ontology of the selected domain is used. The task here is mapping the entities and its relations that extracted from the posed NLQ to concepts ontology. For example, *bank* is the word that needs to be disambiguated; thus, entities and relationship are mapped to the ontology, and each unambiguous entity and relation must match a concept or its synonym and relation in ontology has the same meaning. In this work, NLQ processor looks up to a ternary relation connect these 3 entities; the relation is any morphological derivation of the major relation which is *deposit*. According to the sub-graph in Figure 2, the entities *student*, and *money* with the relation *deposit* are connected through the relation *deposited-in* to the word *bank* that has sense #2 (Financial Institute).

The semantic representation of the NLQ is automatically derived using SRL method and attached to syntactic chunks. As previously mentioned, the semantic representation is represented by the predicate-argument structure, extended with some additional semantic information that assists in answer matching process. For example, Table 2 illustrates how roles assigned to the words of the NLQ.

Table 2: Labeling NLQ's words

Role label	NLQ Words
Pred	Deposit
Arg0	Student
Arg1	Money
Arg-Loc	Bank

Since, the *SRL* is a binary structure, thus, the final semantic representation is derived like *Deposit/P<Student/Arg0,Money/Arg1,Bank.bank#2/Arg-Loc>*.

4.1.3 Answer Type Recognizer

The task here is to classify the NLQ by its expected answer type (EAT). This task is helpful in a process of retrieving answers eventually. Knowing the EAT will impose some constraints on the potential answers. For example, given a NLQ "*When can I register course for the fall semester?*", this NLQ is expected to be classified into an answer type of TIME, which is the only candidate answers that are TIME type need to be considered.

4.2 Answer Processing

Answer processing module is responsible for applying a match method between the extracted NLQ semantic representation and its potential answers. In this work, the potential answers also are normalized to a predicate-argument structure, and then stored in a database. For example, Figure 4 shows how potential answer is processed

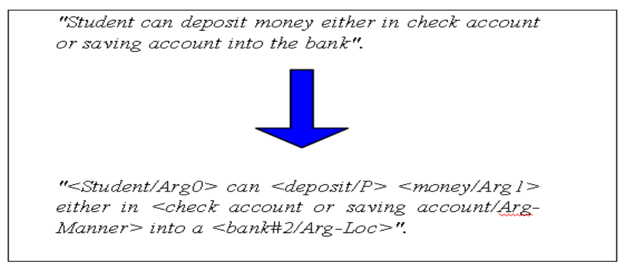

Fig. 4 Semantic role labeling of the potential answer

Since the expected type of the answer is Manner, the matcher must identify that *<saving account or checking account>* is the answer. For this task we built a rule-based matcher.

5. Experiment Evaluation

Evaluation has always been a difficult in WSD research [3, 28, 29]. In this work, we evaluate CKCO approach and its effectiveness in a university QA domain based on in *vivo* evaluation. With this evaluation approach we can tell if our approach is working in the sense of actually improving performance on QA. C# language of programming is used for developing, and widows XP as operating system. There are no data sets concerns with NLQs posed to a university domain. Therefore, in this work, we have collected our own data set which contains 200 natural language questions for the university domain from various universities' websites.

5.1 Experiment Setup and Result

This section presents, the procedures involved in conducting the experiment and the environment of the experiment. In this work, to perform POS a lexicon consists of 400 word has been built. The lexicon contains potential classes of stored words. Classes are categorized into 8 major types which are *noun*, *verb*, *pronoun*, *preposition*, *adverb*, *adjectives*, *article (determiner), and auxiliary verbs*. A linguistic knowledge store is built to assist the chunker in recognizing the phrases in the NLQ. There are three types of phrases considered in this work which are *NP*, *VP*, and *PP*.

A context knowledgebase consists of 300 word is conducted and each individual word is labeled with its senses and potential contexts. The context consists of a set of words which frequently used in a university domain and can indicate to a particular context. Finally, ontology of concepts in a university domain is built. Theses concepts are obtained and represented manually from FAQs (Frequently-Asked Questions) of several websites of universities. A test set includes 200 NLQ which are manually extracted from various universities' websites. This research included five new students to pose NLQs and check if the system can give a response and extracting the correct meaning of the NLQs. Table 3 shows some NLQS that may be posed to this system.

Table 3: Examples of NLQs

NO	Natural language questions (NLQs)
1	How can I deposit money into a bank?
2	How can I cross the river to the west bank?
3	Where do I get my class timetable?
4	Where can I find a bank on campus?
5	How can a student activate his Net account?
6	When can a student join the term?
7	How should use my portfolio?
8	Where can a student buy notes of a lecture?
9	Where does a student check the state of his application?
10	Where can a student find a job in the university?
11	Where can a student find a contact for potential supervisors?
12	Where can a student report a statement about a problem?
13	How does a student submit an application for the college?
14	When can I receive the check of salary?
15	How does a student create an account in the bank?
16	What is the deadline to submit my application?
17	How can a student change his program under the same college?
18	How can I contact a current student in my area?
19	When can I register courses for the fall semester?
20	How do I connect my computer to the Net?

The user may enter NLQs contain ambiguous and unambiguous words, NLQs contain only unambiguous words, or NLQs contain only ambiguous words. The results were calculated based on the number of ambiguous entities that was successfully recognized (AER), the number of correct semantic representations (CSR) for NLQs, and the number of correct answers (CA). Obtained results show that the number of correct semantic representations is same as the number of correct answers. That is because obtaining the correct semantic representation will lead to a correct answer. Sometimes the processor cannot attach the correct semantic to the word. This happens when a NLQ contains more than one ambiguous word which makes it difficult to determine the context of the NLQ.

The notion of precision is used to evaluate the performance of the approach. Precision is an important measure of performance for WSD and QA [29]. Eq 1 illustrates how precision is calculated:

$$\Pr ecision = \frac{CA}{AER} \qquad (1)$$

$$P = \frac{182}{218} = 0.834$$

6. Conclusions and Future Work

In this paper, a novel approach called CKCO for resolving lexical semantic ambiguity in NLQ posed to QA system is proposed. The proposed approach is obtained by combining two pieces of knowledge; context knowledge and ontology of concepts knowledge of interesting domain, into a shallow natural language processing (SNLP) technique. According to the obtained results, it can be concluded that, the proposed approach is capable of resolving ambiguous words in the NLQ which consists of multi words. The significant contribution of this research work is a new technique for resolving lexical ambiguity in NLQ posed to a QA system. In the future, we are looking at resolving the lexical semantic ambiguity in NLQ with different and more complicated structures.

References

[1] U. Hermjakob, “Parsing and Question Classification for Question Answering”, In: Proceedings of the Workshop on Open-Domain Question Answering at ACL, 2001.

[2] J. C. Hung, C. H. Wang, C. Y. Yang, M. S. Chiu, and G. Yee, “Applying Word Sense Disambiguation to Question Answering System for E-Learning”, In Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA’05), Vol. 01, 2005, pp. 157-162.

[3] R. Navigli, “Word sense disambiguation: a survey”, ACM Computing Surveys, Vol. 41, No. 2, 2009, pp. 1–69.

[4] D.S Wang, “A Domain-Specific Question Answering System Based on Ontology and Question Templates”, In Proceedings of the 11th ACIS International Conference on Software Engineering, 2010, pp. 151-156.

[5] E. M. Voorhees, “Overview of the TREC 2003 Question Answering” Track, In: Proceedings of the Twelfth Text REtrieval Conference, 2003.

[6] B. Magnini et al., “Overview of the clef 2004 Multilingual Question Answering Track”, Cross-Language Evaluation Forum (CLEF), 2004,371–391.

[7] J. Fukumoto, T. Kato, and F. Masui, “An evaluation of Question Answering Challenge”, In: (qac-1) at the NTCIR Workshop 3. SIGIR Forum, Vol. 38, No. 1, 2004, pp. 25–28.

[8] V. Guda; S. Sanampudi, and I. Manikyamba, “Approaches for Question Answering Systems”, In International Journal of Engineering Science and Technology (IJEST), Vol. 3, No. 2, 2011, pp. 990-995.

[9] D. Moldovan, S. Harabagiu, M. Pa¸sca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus, “The structure and performance of an open domain question answering system”. In Proceedings ACL, 2000, pp. 563-570.

[10] E. Hovy, H. Hermjakob, M. Junk, and C.-Y. Lin, “Question Answering in Webclopedia”. In Proceedings TREC-9, 2000.

[11] J. Prager, E. Brown, A. Coden, and D. Radev, “Question-Answering by Predicative Annotation”, In Proceedings SIGIR, 2000, pp. 184-191.

[12] J. Burger et al, Issues, “Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)”, Technical report, National Institute of Standards and Technology. 2003, Available on-line at http://wwwnlpir.nist.gov/projects/duc/papers/QA .roadmappaperv2.pdf.

[13] H. M. Alfwareh and S. Jusoh, “Resolving Ambiguous Entity through Context Knowledge and Fuzzy Approach”, International Journal of Computer Science & Engineering, Vol. 3, No. 1, 2011, pp. 410 -422.

[14] Y-H. Lee, et al., “Complex Question Answering with ASQA at NTCIR 7 ACLIA”, In Proceedings of NTCIR-7 Workshop Meeting, Tokyo, Japan, 2008.

[15] G. Hao et al., “Shallow Semantic Analysis of Chinese Question in Book Information Domain”, In Proceedings of the Third International Symposium on Intelligent Information Technology Application, No. 01, 2009.

[16] C. Unger, and P. Cimiano, “Representing and Resolving Ambiguities in Ontology Based Question Answering”, In Proceedings of the TextInfer Workshop on Textual Entailment, EMNLP, 2011, pp. 40-49.

[17] D. Buscaldi, “Resource Integration for Question Answering and Geographical Information Retrieval”, 2007.

[18] P. Chen, W. Ding, M. Choly, C. Bowes, and UHD. Houston, “Word Sense Disambiguation with Automatically Acquired Knowledge”, This paper appears in: Intelligent Systems, IEEE, 2010.

[19] S. Harabagiu, and D. Moldovan, “An Intelligent System for Question Answering”, In Proceedings of the 5th International Conference on Intelligent Systems, 1996, pp. 71–75.

[20] S. Liu, C. Yu, and W. Meng, “Word Sense Disambiguation in Queries”, in CIKM , 2005, pp. 525–532.

[21] J. Liang, T. Nguyen, K. Koperski, and G. Marchisio, “Ontology-Based Natural Language Query Processing for the Biological Domain”, In: Proceeding BioNLP '06 Proceedings of the Workshop on Linking Natural Language Processing and Biology, 2006, pp. 9-16.

[22] Q. Guo, K. Wu, and W. Li, “The Research and Realization about Question Answer System based on Natural Language”, In ICICIC, 2007.

[23] M. Walas and K. Jassem, “Named Entity Recognition in a Polish Question Answering System”, Intelligent Information Systems, 2001, pp. 181–191.

[24] D. Molla and J. L. Vicedo, “Question Answering in Restricted Domains: An Overview, Association for Computational Linguistics”, Vol. 33, No. 1, 2007, pp. 41-61.

[25] M. Aggarwal, “Information Retrieval and Question Answering NLP Approach: An Artificial Intelligence Application”, International Journal of Soft Computing and Engineering (IJSCE) ISSN, Vol. 1 No. 11, 2011, pp. 43-45.

[26] N. Ide, and J. Véronis. “Word Sense Disambiguation: The State of the Art”. Computational Linguistics, Vol. 24, No. 1, 1998, pp. 1-40.

[27] R. Navigali, “Word Sense Disambiguation: A Survey”, In ACM Computing Surveys, Vol. 41, No. 2, 2009, pp. 1-69.

[28] M. Sanderson, “Word Sense Disambiguation and Information Retrieval”, thesis, University of Glasgow, 1996.

[29] D. Jurafsky and J. H. Martin, Speech and Language Processing: An introduction to Natural language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, United States of America, 2009.

[30] S. Jusoh and H.M. AlFawareh, “Natural Language Interface for Online Sale System”, In: proceeding of International Conference on Intelligent & Advanced Systems (ICIAS), 2007, pp. 224-228.

[31] R. Tang, “Using a Machine Learning Approach for Building Natural Language Interfaces for Databases: Application of Advanced Techniques in Inductive Logic Programming,Computer Science”, Informatics & Electrical Engineering, Vol. 2, No. 1, 2008.

[32] D. Lin and P. Pantel, “Discovery of Inference Rules for Question Answering”, Natural Language Engineering, Vol. 7, No. 4, 2001, pp. 343-360.

[33] B. Magnini and G. Cavaglia, "Integrating Subject Field Codes into WordNet", In Proceedings of LREC. 2000, pp. 1413--1418.

Omar Al-Harbi is a PhD student in Islamic Science University of Malaysia (USIM). He earned his Bachelor degree in computer science from Jerash University, Jordan, and his Master degree in IT from Northern University of Malaysia (University Utara Malaysia UUM). His PhD concerns with Question Answering systems, natural language understanding, knowledge engineering, ontology.

Shaidah Jusoh is an associate professor of the Department of Computer Science in the Faculty of Science & Information

Technology at Zarqa University, Jordan. She earned degrees of Master in Computer Science and PhD in Engineering System & Computing from University of Guelph, Canada. Her research interests include, an automated text summarization, question-answering system, online social network, data mining, and information extraction.

Norita Md Norwawi is associate professor. She is the head of program for computer science at the faculty of science and technology, Universiti Sains Islam Malaysia. She graduated with A BSc in computer science at University of New South Wales, Australia in 1987. In 1994 she obtained her MSc in Computer science from Universiti Kebangsaan Malaysia. She was conferred with a PhD in 2004 from Universiti Utara Malaysia specializing in the area of Artificial Intelligence. She has published her work in peer reviewed journals and international proceedings. Her research interest is in the application of AI techniques especially in information security and assurance such as data mining, multiagent system and visualization.