Design of Ontology-based Question Answering System for Incompleted Sentence Problem

Rajif Agung Yunmar, I Wayan Wprayoga Wisesa
Informatics Engineering, Institut Teknologi Sumatera, Lampung Selatan, ID

Abstract. Ontology-based Question Answering System (QAS) has advantage on representing semantic knowledge concept of an object, property, and relation between objects on a particular domain. Ontology-based QAS requires a proper definition of objects, properties, and their relationships in order to work correctly. It is very challenging topic for Bahasa Indonesia, especially when dealing with incomplete question sentences. For example: the sentence with no question word or no question object, or no obscure adverb question. This study designed an Ontology-based QAS that capable for dealing incomplete question sentence. Two approach method has been done to deal with the problem: First, designed an ontology that used a keyword property to find the question objective; Second, designed an architecture that transform the question in natural language into instruction that can be understood by computer. Experiment results shows that the ontology design plays an important role for Ontology-based QAS as well as system design.

1. Introduction
Nowadays, looking for an information is becoming easier. Information can be found from many sources, both electronic and conventional. Website as electronic media provides thousands of pages just for one or similar information. Actually, these conveniences makes users difficult to find appropriate information.

Question Answering System (QAS) is a system that allows users to request the information in the form of questions that written in natural language, then QAS return a short text or sentence phrase as the answer [1]. Thus, users do not need to bother seeking one by one web pages to find the information needed.

Ontology-based QAS uses SPARQL queries to generate information and answer questions that asked by users. SPARQL queries can answer a question if the objects, relations, and properties that involved in a problem are precisely defined [2], [3]. But, in Bahasa Indonesia there are several cases that allow to use an incomplete question sentences. For example: the question that not contained question word, no question object, or no definitely question adverb. Kind of sentences can make objects, predicates, and properties that needed by SPARQL queries can be undefined, thus the information that generated by SPARQL queries becomes incorrect.

Based on the problem background, this study attempts to design an Ontology-based QAS that can answer the questions requested by users in Bahasa Indonesia correctly, according to question context, even though with incomplete question sentences input.
2. Literature Study

2.1. Question Answering System (QAS)
QAS is one solution to the number of user that requests to get information quickly and accurately [4]. The biggest challenge of a QAS is how to classify the questions into particular category, then find the right answer from the document or knowledge base that QAS owned [5]. QAS becomes an alternative to get information at this time, where search results of search engines do not always match with the user needed [6].

2.2. Question Sentence
Question sentence is a sentence that arranged by certain question words. For example: what, where, when, which, how, etc. But in Bahasa Indonesia (Indonesian language), it is also possible to ask questions that do not use question words, no question objects, or no adverb. Based on the question purpose, both verbally and in writing, in Bahasa Indonesia question sentences can be divided into several types, namely [7]:
- Ordinary question sentence
  The ordinary question sentence used to aim of digging up information or further explanation regarding a problem. This question sentence usually used in interviews or dialogues.
- Rhetorical question sentence
  The retorical question sentence is a question sentence that does not require an answer. This question sentence tends to aim for attention or intends to provide enthusiasm, objection, or criticism. This question sentence usually used in speeches.
- Clarification question sentence
  The clarification question sentence also called confirmation question sentence. This question sentence is used to get clarification or confirmation of a problem. The answer that will be given is yes or no.
- Disguised question sentence
  The disguised question sentence is a question that is not intended to explore information, clarification, or confirmation, but it is used for other purposes. For example: command sentences, persuade, insinuating, etc.

2.3. Ontology
Ontology is a model from the real world on a particular domain that contains relationships between concepts, both hierarchically and relations. The purpose of using ontology is to improve text processing automation by providing language concept representations that are independent, and meaning-based [8]. Ontology is chosen as the knowledge base because it is able to represent knowledge information based on the semantic concept of an object, property, and relations between objects that occur in a particular domain [9]. In ontology, knowledge relation represented in Subject-Predicate-Objects form. Interestingly, objects in that relation can be subject on another relation, as seen in Figure 2.1. The use of ontology is expected to produce a system with knowledge which not only can be understood by humans, but can also be understood by machines. Thus the information that provided by the system that built on it can be more relevant [10].

2.4. SPARQL
SPARQL (SPARQL Protocol and RDF Query Language) is a semantic query language used to process data that stored on the ontology knowledge base in RDF (Resaource Description Framework) format [12], [13]. The language that used by SPARQL is similar to SQL in the relational database. SPARQL makes it possible to perform complex join operations in different databases in a simple way, turning RDF data into other vocabularies, etc. [14].
3. Related Works

Several studies related to the development of QAS with unclear questions have been made, including study of Fukumoto’s, et al. which uses clue words that extracted from documents using Named Entity Recognition modules. The use of clue word aims to produce appropriate information documents by reducing candidate answers based on clue words that appear [15].

Shekarpour, et al. developed the SPARQL query generator to solve the problem of simplifying information search on the DBPedia ontology graph so can be used by lay users. This study calculates the suitability of IRI (Internationalized Resource Identifier) candidates for each keyword given by the user, then use the filtered IRI and has valid status to generate SPARQL queries [3].

Aunimo, et al. discuss the development of QAS for incomplete and noisy data. The system built in client-server architecture. The clients built with CGI in charge to collect from SMS then send it to server to find the answer. This study discuss the contents of SMS messages which are usually limited, whereas character that can be accommodated maximum 160 character, and often contains various abbreviations. To get results, the server looks for the most suitable question in the relational database, which uses a vector-based method to represent the string of questions and compare it with the cosine similarity measure [16].

The studies that mentioned above does not clearly explain how a QAS can answer a question if the input of the question sentence is incomplete, especially in Bahasa Indonesia. The other researchers also developed ontology-based QAS, but there are not discuss how the system could answer the incomplete sentences, namely: Lopez, et al. [17], Devi and Dua [18], Syarief [19], and so on.

4. Methodology

This research designs ontology-based QAS that can answer incomplete question sentences. For example: there are not containing question word, there are no question objects, or there is no question adverb. While the types of question sentences that will be discussed are ordinary question sentences in Bahasa Indonesia. The problem solves in two ways: first, designed an ontology that used a keyword property to find the question objective; second, designed system architecture that used to process questions in natural language into instruction that can be understood by the computer, so the questions asked can be answered.
4.1. Ontology Design
Besides representing the knowledge, ontology structure that will be designed in this study also plays an
important role in helping to solve the QAS problem against incomplete question sentences. Ontology
built consists of four parts, namely: class, instance, predicate, and property. Figure 4.1 shows the
ontology design in College Student Admissions domain, while Table 4.1 shows an example case of a
class design on that ontology domain.

Figure 4.1 Ontology design in College Student Admissions domain.

Table 4.1 The example class design on College Student Admissions ontology domain

| Class            | Property                  | Class            | Property                  |
|------------------|---------------------------|------------------|---------------------------|
| Beasiswa         | beasiswa_nama             | Kegiatan Mahasiswa | kegiatan_nama             |
|                  | beasiswa_prosedur         |                  | kegiatan_jadwal           |
|                  | beasiswa_syarat           |                  | kegiatan_prosedur         |
|                  | beasiswa_keterangan       |                  | kegiatan_keterangan       |
|                  | keyword                   |                  | keyword                   |
| Biaya Kuliah     | ukt_golongan              | Fasilitas        | fasilitas_nama            |
|                  | ukt_nominal               |                  | fasilitas_keterangan      |
|                  | keyword                   |                  | keyword                   |
| Program Studi    | prodi_nama                | Jalur Masuk      | jalurmasuk_nama           |
|                  | prodi_kuota               |                  | jalurmasuk_jadwal         |
|                  | prodi_akreditasi          |                  | jalurmasuk_keterangan     |
|                  | keyword                   |                  | keyword                   |

The ontology design in Table 4.1 can solve problems that discuss in this study by creating properties
named *keyword* that are always attached to all existing classes. Through the keyword property, it can
be determined which class or instance object thats meant in a user's question. The examples of keyword
implementation in instance class can be seen in Table 4.2. Each class instance can contained many
keyword properties, this useful to facilitate various keywords that refer to one meaning or same term.
4.2. System Design
This section is a stage where questions that entered by users in natural language form will be changed into instructions that can be understood by the computer. Some steps that will be passed include: Stemming, Stopword Remover, Tokenizing, Post Tagging, Keyword Identification, and SPARQL Query Formation. The proposed system that designed for analyze question sentence can be seen in Figure 4.2.

A. Stemming
The stemming process used to eliminate the affix in a word so the basic word can be obtained. In this study, Nazief & Adriani stemming algorithm used to remove the affix. According to Ledy, this algorithm provides better results than other stemming algorithms, especially stemming in Bahasa Indonesia [20].

B. Stopword Remover
The stopword remover process used to eliminate unnecessary words, or words that have no meaning. For the example unnecessary words in Bahasa Indonesia namely: di, yang, pada, ke, and so forth. This process is done by comparing each word in question sentence with stopwords dictionary. If the word that tested matches with the word in the dictionary stopwords, the word will be deleted.

C. Tokenizing, Post Tagging, dan Keyword Identification
The Tokenizing process will change the question sentence to lowercase, delete characters and symbols that are not needed. Furthermore, the Post Tagging process will split the sentences into the words and classify that words based on its type. For example: verbs, nouns, adjectives, etc. Post Tagging stage also identify the words that might be keywords.

| Class                       | Class Instance | Property                      | Value                                      |
|-----------------------------|----------------|-------------------------------|--------------------------------------------|
| Biaya Kuliah                | UKT_1          | uktt_golongan                 | Uang Kuliah Tunggal Golongan 1            |
|                             |                | uktt_nominal                  | Rp. 500.000.-                             |
|                             |                | keyword                       | UKT                                        |
|                             |                | keyword                       | UKT 1                                      |
| Biaya Kuliah                | UKT_2          | uktt_golongan                 | Uang Kuliah Tunggal Golongan 2            |
|                             |                | uktt_nominal                  | Rp. 1.000.000.-                            |
|                             |                | keyword                       | UKT                                        |
|                             |                | keyword                       | UKT 2                                      |
| Program Studi               | PS_Teknik_Informatika | prodi_nama                   | Teknik Informatika                        |
|                             |                | prodi_kuota                   | 180                                        |
|                             |                | prodi_akreditasi              | C                                          |
|                             |                | keyword                       | Program Studi                              |
|                             |                | keyword                       | Informatika                                |
|                             |                | keyword                       | Teknik Informatika                         |
|                             |                | keyword                       | IF                                         |
| Program Studi               | PS_Teknik_Gematika | prodi_nama                   | Teknik Geomatika                           |
|                             |                | prodi_kuota                   | 180                                        |
|                             |                | prodi_akreditasi              | C                                          |
|                             |                | keyword                       | Program Studi                              |
|                             |                | keyword                       | Geomatika                                  |
|                             |                | keyword                       | Teknik Geomatika                           |
|                             |                | keyword                       | GT                                         |
Figure 4.2 System design for analyze question sentence.

D. SPARQL Query Formation
There are three processes that conducted in SPARQL Query Formation stage, namely: keyword association, predicate identification, and property identification. These three processes related with how the SPARQL queries built, which used to draw answers from the ontology knowledge base. Figure 4.3 describes the SPARQL Query Formation process.

Figure 4.3 The process of SPARQL query formation.

The first process that conducted this stage is the keyword association. All keywords that identified at the Post Tagging stage have an important role in SPARQL queries formation. Keywords that have been identified will be associated with certain instance. This association can be carried because each instance in ontology has been labeled with certain keywords, as can seen in example that list in Table 4.2.
The second process of SPARQL query formation is to identify the predicates based on the keyword that associated in the first stage. The predicate identification is conducted if there are more than one keyword found. This process also determines which keywords will be the subject, and which keywords become objects. Predicate search is performed using the SPARQL query formula as shown in Figure 4.4. The results of these queries will be filtered by comparing the keywords with existing subjects and objects.

![Figure 4.4 SPARQL query formula to find the predicates.](image)

The next process at the stage of SPARQL query formation is property identification. The property identification process is important to do, because it is related to what kind of information will be displayed as the answer of a question. For example, instance of “Verifikasi Berkas Fisik” have several properties, including: nama kegiatan (activity name), syarat (terms), prosedur (procedures), jadwal (schedule), etc. If the user needs to know the information about how to follow the “Verifikasi Berkas Fisik” activity, then the properties that will be displayed is procedures (procedures). Property identification process is conducted by associating a word with a specific purpose. This word association uses an Property Association Table that specifically used to mapping the word to certain property. For example: the word “kapan” (when) will be associated with “jadwal” (schedule). Another example of word association that can be related to an instance property can be seen in Property Association Table that listed in Table 4.3.

| Word   | Property  | Word   | Property  |
|--------|-----------|--------|-----------|
| apa    | nama      | dimana | tempat    |
| bagaimana | prosedur | kapan  | jadwal    |
| berapa | nominal, jumlah, kuota | tanggal | jadwal |
| biaya  | nominal   | pengumuman | keterangan |

![Figure 4.5 (a) Template of SPARQL query formation template with one keyword identified. (b) Template of SPARQL query formation template with two keyword identified.](image)
After getting the keyword, subject, object, predicate, and property, the last process at this stage is the formulating the SPARQL queries using the specified template. Among the SPARQL query templates that used in this study can seen in Figure 4.5.

4.3. SPARQL Execution

After the SPARQL query is formed, the next step is to execute the query. There are several tools that can be used, but in this study SPARQL queries were executed with Apache Jena Fuseki software. The execution of SPARQL query generates text that is used as the answer to the question that posed by the user.

5. Discussion

This section will conduct a trial experiment in cases of question sentences towards the proposed ontology and system design. The test case were conducted on three groups of question sentences, namely: complete question sentences, incomplete question sentences without question objects, and incomplete question sentences without question words and questioning objects. The test case scenario can be seen in Table 5.1.

| Q | W | S | O | Question | Identification results based on ontology and system design approach | Answer Value |
|---|---|---|---|---------|-------------------------------------------------|-------------|
| √ | √ | √ | √ | Prodi apa saja yang ada di ITERA? (intent: what major exist at ITERA?) | prodi - - - | True |
| √ | √ | √ | √ | Berapa biaya kuliah di ITERA? (intent: how much tuition fee at ITERA?) | biaya kuliah - - nomial ukt | True |
| √ | √ | √ | √ | Hasil seleksi bidikmisi itera bagi yang lolos snmptn kapan? (intent: when does the selection result of Bidikmisi scholarship for student who passed SNMPTN exam announced?) | bidikmisi, snmptn bidikmisi, snmptn - - | False |
| √ | √ | √ | √ | Prodi apa saja yang ada di Jurusan Sains? (intent: what study program exists at the faculty of natural science ITERA?) | prodi, jurusan sains prodi, jurusan sains bagianFakultas nama prodi | True |
| √ | √ | - | - | Kapan pengumuman UKT? (intent: when does the decision of tuition fee grades announced?) | uk - - | jadwal True |
| √ | √ | - | - | Kapan diumumkan hasil seleksi ujian masuk jurur prestasi khusus ya? (intent: when does the result of special track exam announced?) | jalur prestasi jalur prestasi - | jadwal True |
| √ | √ | - | - | Apa syarat banding UKT? (intent: what are the terms for tuition fee appeal?) | banding uk - - | syarat True |
| √ | √ | - | - | Berapa biaya kuliah di Teknik Informatika? (intent: how much the tuition fee at Informatics Engineering department?) | biaya kuliah, teknik informatika ukt, teknik informatika punyaBiaya | nominal ukt True |
| - | √ | - | - | Info mengenai bidikmisi, dong? (intent: may I asking for the information about Bidikmisi scholarship?) | bidikmisi bidikmisi - | keterangan True |
The test case that contained in Table 5.1 shows the question sentence can be identified through the keywords. The test case shows that the completeness of a question sentence, the present of question word (QW), subject (S), or object (O) does not affect the answer, provided that the keyword exists in question sentence. On the other case, the questions that are not containing instance, predicate, or property also can still be answered, provided that the keyword exists in question sentence.

The test case in Table 5.1 shows a question sentence in which can found the keyword, and the instance but produces the wrong answer (False). Analysis of the case is the absence of a predicate in the ontology design that connects the instance, that is predicate between bidikmisi and snmptn. Which bidikmisi is instance of Beasiswa, and snmptn is the intance of JalurMasuk class. As can seen in Figure 4.1 that both of Beasiswa and JalurMasuk has no relation. While the test case at Table 5.1 shows that all question sentences in which there are two instance found, it is certain to have one predicate that connects between the two instance.

6. Conclusion
Based on the discussion above, it can be concluded that the keyword plays an important role in answering a question especially in Bahasa Indonesia, even though the question does not have a question word (QW), subject (S), or object (O). Another conclusion, it is important in an ontology design to relate two instances that should have linked using a predicate. It is intended so drawing the answers that related with two instance (that should be connected) becomes valid (True).

The challenge for future research is how to answer a question in which there are containing more than two instance. To answer these challenges, this study has provided a knowledge of the importance of keywords and predicates in the design of an ontology in the QAS domain.

7. References
[1] C. Monz, “From Document Retrieval to Question Answering,” Universiteit van Amsterdam, 2003.
[2] M. Arenas and J. Pérez, “Querying Semantic Web Data with SPARQL,” in Proceedings of the 30th symposium on Principles of database systems of data - PODS ’11, 2011, pp. 305–316.
[3] S. Shekarpour, A. N. Ngomo, and D. Gerber, “Keyword-driven SPARQL Query Generation Leveraging Background Knowledge,” in Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 2013–210.
[4] A. Tahri and O. Tibermacine, “DBPedia based factoid question answering system,” Int. J. Web
[5] H. Toba and M. Adriani, “Pattern Based Approach in Indonesian Question-Answering System,” Fac. Comput. Sci. Univ. Indonesia., pp. 1–8, 1999.

[6] F. Baskoro and W. Rubikartin, “Pengembangan Question/Answering Portal dengan Answer Quality Predictor pada Aplikasi e-Learning,” in Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009), 2009.

[7] M. Irman, T. W. Prastowo, and Nurdin, Bahasa Indonesia 2 Untuk SMK/MAK Semua Program Keahlian. Jakarta: Pusat Perbukuan Departemen Pendidikan Nasional, 2008.

[8] R. Samo, Y. Anistyasari, and R. Fitri, Semantic Search. Pencarian Berdasarkan Konten. Yogyakarta: Penerbit ANDI, 2012.

[9] Azhari, Subanar, R. Wardoyo, and S. Hartati, “Model Representasi Informasi Dan Pengetahuan Untuk Proyek-Proyek Perusahaan Dengan,” J. Ilm. Teknol. Inf., vol. 7, pp. 85–92, 2008.

[10] S. Lee, P. Ryu, and K. Choi, “Ontology-based Question Answering System QA Processing Unit : Triplet,” in The 6th International Semantic Web Conference, 2007.

[11] F. L. Constantin, “How to tune this ontology for simple geometric shape recognition?,” 2012. [Online]. Available: https://stackoverflow.com/questions/13892848/how-to-tune-this-ontology-for-simple-geometric-shape-recognition. [Accessed: 16-Sep-2018].

[12] G. Antoniou, P. Groth, F. van Harmelen, and R. Hoekstra, A Semantic Web Primer third edition. Cambridge, Massachusetts: The MIT Press, 2012.

[13] E. Prud’hommeaux and Andy Seaborne, “SPARQL Query Language for RDF,” 2013. [Online]. Available: https://www.w3.org/TR/rdf-sparql-query/.

[14] L. Afuan and A. SN, “Penerapan Semantik Web pada Ontologi Learning Resource Repositori,” in Seminar Nasional Teknologi Informasi dan Komunikasi 2016 (SENTIKA 2016), 2016, pp. 18–19.

[15] J. Fukumoto, N. Aburai, and R. Yamanishi, “Interactive Document Expansion for Answer Extraction of Question Answering System,” Procedia - Procedia Comput. Sci., vol. 22, pp. 991–1000, 2013.

[16] L. Aunimo, O. Heinonen, R. Kuuskoski, J. Makkonen, R. Petit, and O. Virtanen, “Question Answering System for Incomplete and Noisy Data,” Eur. Conf. Inf. Retr., pp. 193–206, 2003.

[17] V. Lopez, V. Uren, E. Motta, and M. Pasin, “AquaLog : An ontology-driven question answering system for organizational semantic intranets.”

[18] M. Devi and M. Dua, “ADANS : An Agriculture Domain Question Answering System using Ontologies,” in International Conference on Computing, Communication and Automation (ICCCA2017), 2017, pp. 122–127.

[19] M. Syarief, “Ontomotif : Ontologi Pencarian Informasi Kendaraan Bermotor,” in Seminar Nasional Sains dan Teknologi 2015 Fakultas Teknik Universitas Muhammadiyah Jakarta, 2015, no. November, pp. 1–10.

[20] L. Agusta, “Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia,” Konf. Nas. Sist. dan Inform. 2009, no. KNS&I09-036, pp. 196–201, 2009.