Extending an Indonesian Semantic Analysis-based Question Answering System with Linguistic and World Knowledge Axioms*

Rahmad Mahendra[a], Septina Dian Larasati[a], and Ruli Manurung[a]

[a]Faculty of Computer Science, University of Indonesia, Depok 16424, Indonesia
rama42@ui.edu, septina.larasati@gmail.com, maruli@cs.ui.ac.id

Abstract. We adopt a previously developed model of deep syntactic and semantic processing to support question answering for Bahasa Indonesia, and extend it by adding a number of axioms designed to encode useful knowledge for answering questions, thus increasing the inferential power of the QA system. We believe this approach can increase the robustness of semantic analysis-based QA systems, whilst simultaneously lightening the burden of complexity in designing semantic attachment rules that transduce logical forms from syntactic structures. We show how these added axioms enable the system to answer questions which previously could not have been answered.

Keywords: bahasa Indonesia, knowledge representation, QA systems, semantic analysis.

1. Introduction

A question anwering (QA) system seeks to provide answers to questions expressed in natural language, where the answers are to be found in a given collection of documents. QA systems typically require more sophisticated linguistic analysis than conventional information retrieval, as they need to reason about various other factors, among others the types of questions, predicate argument structure, and result aggregation.

In the work presented in this paper, we start from a unification-based grammar augmented with lambda-calculus rules that constructs semantic representations of Indonesian declarative sentences. To these representations we subsequently combine a suite of axioms designed to encode linguistic and world knowledge, and assert them into a knowledge base. A separate QA module answers queries by unifying the question semantic representation with the augmented set knowledge base.

In Sections 2 and 3 we first discuss some relevant past work. Section 4 presents the overall framework of our system, and Section 5 discusses the semantic representation underlying our approach, arguing for some form of axiomatic post-processing. Finally, Sections 6 and 7 present the axioms themselves, along with some examples of how they contribute to the QA process.

2. Lightweight semantic approaches to Question Answering

In general, there are two approaches to QA: the bottom-up approach employs "shallow" statistical methods such as keyword-based retrieval, which benefit from the sheer size of large electronic collections of documents nowadays available (e.g. the web), and are very robust. Unfortunately, these probabilistic methods are sometimes unable to perform the required inference for answering complex questions. On the other hand, the top-down approach uses "deeper" linguistic methods to obtain semantic representations of both the question and (a subset of) documents. The resulting logical forms enable precise identification of answers, sometimes in cases where they are not explicitly stated in the source documents. However, these

* Copyright 2008 by Rahmad Mahendra, Septina Dian Larasati, and Ruli Manurung

deep methods typically require carefully-engineered, language-specific resources that are very costly to produce and not very robust.

More recently, work has been done in developing QA systems that try to combine the two approaches, e.g. (Moldovan et al., 2003), (Narayanan and Harabagiu, 2004), and (Shen and Lapata, 2007). Crucially, these systems capture predicate argument structure that is shown to be essential for complex question answering. Additionally, semantic representations enable logical inference, allowing the QA systems to answer more complex queries by exploiting knowledge encoded in ontologies such as WordNet, SUMO, and various other Semantic Web-based resources.

COGEX, the system reported in (Moldovan et al., 2003), is a QA system that employs a robust syntactic parser that essentially outputs a quasi-logical form containing part of speech information and general predicate argument structure. This output is passed to a theorem prover, and question answering is modelled as a theorem proving task. To aid this process, several axioms are added: NLP axioms establish the semantic content ignored by the robust parser from syntactic constructions such as complex nominals, coordinated conjunctions, appositions, and possesives. World knowledge axioms augment the knowledge extracted from the document collection with knowledge from existing ontologies, e.g. WordNet (Fellbaum, 1998).

3. Question answering in bahasa Indonesia

Bahasa Indonesia (hereinafter simply 'Indonesian') is the official language of Indonesia, spoken by over 100 million people. Given this fact, we believe it is underrepresented in terms of research into Indonesian QA, and Indonesian NLP in general.

There has been some work on developing QA systems for Indonesian. (Wijono et al., 2006) sought to achieve multilingual QA by answering queries in Indonesian based on English documents. Questions are classified based on a manually constructed taxonomy of Indonesian questions. The query is then automatically translated into English using a commercial translator available online[1], and from then on is handled as a purely English QA task. (Purwarianti et al., 2007) uses a machine learning method to develop the question and answer classifier modules based on a corpus of raw text.

(Larasati and Manurung, 2007) presented a purely symbolic approach that adopts a deeper linguistic approach, leveraging a previously built syntactic parser for the Indonesian language (Joice, 2002). We adopt this approach and extend it with some post-processing of the semantic representations with a suite of axioms.

4. Our QA system framework

The overall framework of our Indonesian QA system consists of the following modules: a syntactic parser, a semantic analyser, and a question answering module augmented with axioms. Following (Larasati and Manurung, 2007), we use a unification-based grammar implemented as a set of DCG rules in Prolog. Since wide coverage is currently not the main aim of our research, we developed a relatively small yet usable handcrafted grammar and lexicon based on the official Indonesian grammar (Alwi et al., 1998).

The semantic analyser module transduces semantic representations from parse trees. These semantic representations are designed to abstract away syntactic variations, allowing sophisticated automated processing of Indonesian texts. We adopt a 'flat' semantic representation (Hobbs, 1985). Details of the specific representation we use is presented in Section 5.1. Adopting the well-known rule-to-rule hypothesis, we augmented the lexicon with semantic information (Section 5.2), and developed semantic attachment rules for each grammar rule (Section 5.3).

Although the above Indonesian semantic analyser is intended to be general-purpose, we have a specific concrete aim of developing a question answering system for Indonesian. Currently, we have implemented a prototype query processor in Prolog. The semantic representations of

[1] http://www.toggletext.com

Indonesian declarative sentences, i.e. as found within a collection of documents, are stored in a clausal knowledge base. Subsequently, the semantic representation of queries are transformed into Prolog rules which, when unified with the clause database, yields the appropriate answer.

5. Semantic representation

In this section we present all the details concerning the semantic representations of Indonesian sentences, i.e. the syntax of logical expressions, the content of lexical semantics, and how the semantic attachment rules are defined and applied.

5.1. Logical expressions

As mentioned above, we adopt a simple 'flat' semantic representation (Hobbs, 1985), where a logical expression is a conjunction of first order logic literals. The arguments of these literals represent domain concepts such as objects and events, while the functors state relations between these concepts. All variables are existentially quantified with the widest possible scope.

Additionally, following the approach in (van Durme et al., 2003), literals are divided into two categories, extrinsic and intrinsic literals. An extrinsic literal defines a relationship between two variables, whereas an intrinsic literal defines a relationship between a variable and its referent as being some semantic concept in some underlying ontology. Examples of intrinsic literals are λX *event(X,Y)*, where *X* is event object *Y*, λX *object(X,Y)*, where *X* is inanimate object *Y*, and λX *location(X,Y)*, where *X* is location object *Y*. Examples of extrinsic literals are $\lambda X \lambda Y$ *agent(X,Y)*, where *X* is the agent of *Y*, and $\lambda X \lambda Y$ *patient(X,Y)*, where *X* is the patient of *Y*. Both types of literals are stored within the lexical semantics entries of the words that convey their meaning, which specify the *Y* variable (see Section 5.2).

Our semantic representation falls into the category of so called neo-Davidsonian approaches, where intrinsic literals are predicates over objects and events, and arguments and modifiers are specified via the thematic relations specified by the extrinsic literals.

5.2. Lexical semantics

Lexical entries of open class words are associated with exactly one intrinsic literal which asserts a reference to the domain concept the word is 'about'. We arbitrarily choose the root form of a synonym to act as the conceptual symbol. Additionally, words may also be associated with extrinsic literals representing thematic relations that must be specified by complements within its syntactic projection.

For instance, the transitive verb *"memakan"* (to eat) has the following lexical semantic representation:

$$\lambda E \lambda A \lambda P \; event(E,memakan) \wedge agent(E,A) \wedge patient(E,P).$$

where *event(E,memakan)* is the intrinsic literal specifying the domain concept, i.e. eating event, and *agent(E,A)* and *patient(E,P)* are extrinsic literals whose variables will be subsequently bound with the subject and object variables through the lambda calculus operation of β-reduction (see Section 5.3 below).

In (Alwi et al. 1998), there are several subcategories of nominals, e.g. temporal, location, object, person, etc. The lexical semantics of nominals is simply the appropriate intrinsic literal, e.g. the semantics of *"dapur"* (kitchen) is λX *location(X,dapur)* and the semantics of "*ayah*" (father) is λX *person(X,ayah).*

Adjunct modifiers such as adjectives and adverbials are associated with a logical expression containing the appropriate intrinsic literal coupled with an extrinsic literal that specifies the thematic relation between the modifier and its head. For example, the semantics of "*indah*" (beautiful) is $\lambda A \; \lambda T$ *property(A,indah)* $\wedge$ *attrib(T,A)*.

The lexical semantics of prepositions and words which coordinate and/or subordinate other clauses is simply the appropriate extrinsic literal which specifies the relation between the

prepositional phrase and its head or the clauses being coordinated. For example, the semantics of "*karena*" (because) is $\lambda X\ \lambda Y\ cause(X,Y)$.

Question words, i.e. *wh* words in Indonesian, e.g. "*apa*" (what), "*siapa*" (who), "*mana*" (mana), are associated with a logical expression that contains two literals. The first is the appropriate literal which would typically be associated with the answer, but instead of specifying the domain concept as the second argument, it is given a variable *Ans*. The second literal is a special *ans(Ans)* literal that indicates a question that is to be processed by the question answering module. For example the lexical semantics of "*siapa"* is $\lambda X\ person(X,Ans) \wedge ans(Ans)$.

Finally, there are several special cases of lexical semantics where morphological processes introduce literals. For example, the suffix "*nya*" amounts to a possessive pronoun, requiring the addition of the literals *person(O,owner)* and *possess(O,X)* to the lexical semantics. For example, we assume that the lexical semantics for the word "*bukunya"* (his/her book) is $\lambda X\ object(X,buku) \wedge person(O,owner) \wedge possess(O,X)$.

5.3. Semantic attachment rules

Frege's principle of compositionality of semantics states that the meaning of a complex expression is determined by the meanings of its parts, and the way in which those parts are combined. In linguistic terms, rules that determine semantic interpretation are defined on the syntactic rules and structures. As a result, we develop *semantic attachment rules* for each syntactic rule in our grammar.

These semantic attachment rules define how the lexical semantics of the constituent words are combined, and in particular how the correct predicate-argument structure is specified. The most common approach is to use *lambda calculus* notation, where predicate-argument structure is controlled through the operation of β-reduction. See (Jurafsky and Martin, 2000) for a clear discussion of this approach (note that they call the process lambda-reduction).

To see an example of the semantic attachment rules and how they are combined, observe the following example, which constructs the semantic representation of the simple declarative sentence "*Ayah memakan nasi"* (father eats rice).

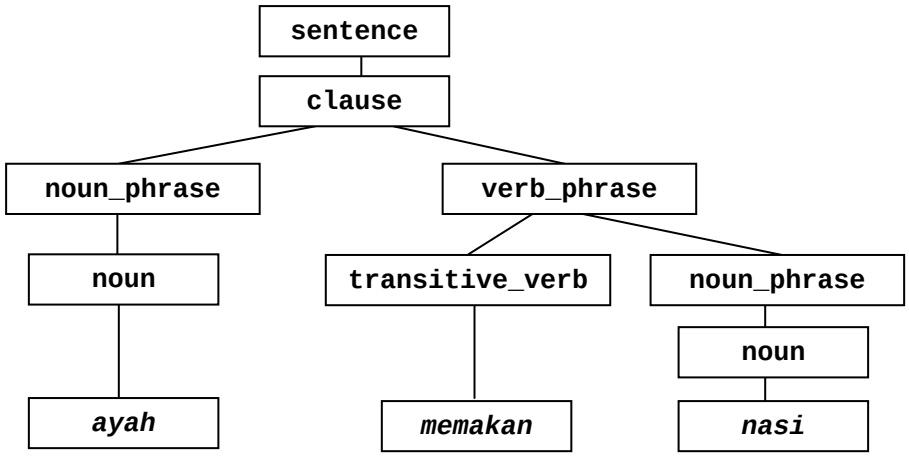

Figure 1: Parse tree for sample sentence *"Ayah memakan nasi"*.

Figure 1 shows how the sentence is parsed by our grammar: a **`sentence`** can consist of a single **`clause`**, which in turn expands to **`noun_phrase`** and **`verb_phrase`**. The **`noun_phrase`** category simply consists of a single **`noun`** lexeme, whereas a **`verb_phrase`** consists of a **`transitive_verb`** lexeme and another **`noun_phrase`** category.

Rules (1)-(3) below show the required syntax rules and corresponding semantic attachment rules, whereas rules (4)-(6) show the lexical semantics entries (see Section 5.2 for discussion of these values):

1. **clause -> noun_phrase, verb_phrase**
   λM (noun_phrase.sem(K) ^ verb_phrase.sem(M)(K))
2. **noun_phrase -> noun**
   noun.sem
3. **verb_phrase -> transitive_verb, noun_phrase**
   λHλG (transitive_verb.sem (H)(N)(G) ^ noun_phrase.sem (N))
4. **noun -> [ayah]**
   λD person(D, ayah)
5. **transitive_verb -> [memakan]**
   λEλPλA (event(E, memakan) ^ agent(E, A) ^ patient(E, P))
6. **noun -> [nasi]**
   λS object(S, nasi)

The ***.sem*** operator indicates the logical expression of the indicated syntactic category. The β-reduction proceeds as follows:

1. Lexical semantics are copied over to the **noun_phrase** categories:
   (2) & (4): ***noun_phrase.sem = λD person(D, ayah)***
   (2) & (6): ***noun_phrase.sem = λS object(S, nasi)***
2. At the **verb_phrase** rule, the semantics **transitive_verb** and **noun_phrase** are substituted and reduced:
   (3), (5) & (2): ***verb_phrase.sem = λHλG (λEλPλA (event(E, memakan) ∧ agent(E, A) ∧ patient(E, P))(H)(N)(G) ∧ λS object(S, nasi)(N))***
   reduces to
   λHλG (event(G, memakan) ∧ agent(G, H) ∧ patient(G, N) ∧ object(N, nasi))
3. At the **clause** rule, the semantics **noun_phrase** and **verb_phrase** are substituted and reduced:
   (1), (2) & (3): ***clause.sem = λM (λD person(D, ayah)(K) ∧ λHλG (event(G, memakan) ∧ agent(G, H) ∧ patient(G, N) ∧ object(N, nasi))(M)(K))***
   reduces to
   λM (person(K, ayah) ∧ event(M, memakan) ∧ agent(M, K) ∧ patient(M, N) ∧ object(N, nasi))

The semantic representation of **sentence**, the sentence, is simply the semantics of the **clause** as shown above.

5.4. The problem with syntax-driven semantic analysis

The complex machinery described in the last three subsections essentially plays one role: to abstract away the syntactic variations from paraphrases that essentially convey the same thing. The semantic representations produced from these paraphrases should be a single canonical representation. Due to the complex nature of natural language, however, this is an extremely complicated task, and often fails to scale up to large collections of text.

For example, in the case of possessives, we identified five different representations produced by the syntax-driven semantic analysis. The noun phrase

raket Rahma
racket Rahma
"Rahma's racket"

yields the semantic representation *λX object(X,raket) ∧ person(A,rahma) ∧ possess(A,X)*, whereas the sentence

Rahma memiliki raket
Rahma owns racket

"Rahma owns a racket"

yields the semantic representation *event(E,memiliki)* ∧ *agent(E,A)* ∧ *patient* ∧ *(E,X)* ∧ *object* ∧ *(X,raket)* ∧ *person(A,rahma).*

One would hope the two semantic representations form an entailment relationship despite the fact the former focuses on the object whereas the latter focuses on the ownership event. Although theoretically we could reformulate the semantic attachment rules to produce a canonical form, we believe these rules are still too closely mapped to the syntactic structure, and thus not yet at a high enough level of abstraction to establish semantic equivalence. Following the approach in (Moldovan et al., 2003), this task is handled by introducing logical axioms as a form of "post-processing". We argue there are two benefits to this approach. Firstly, it reduces the burden on the design of the semantic attachment rules having to produce canonical forms, i.e. syntactic variations may still be present. This in turn enables the use of wider-coverage grammars. Secondly, it allows us to encode external knowledge not available in the original document collection. The following subsection discusses the axioms we have designed and implemented.

6. Axioms

To refine the semantic representation produced by the previous semantic analysis module, we build a post-processing semantic analysis by defining axioms and adding it to the system. These axioms broadly fall into two categories, NLP axioms and world knowledge axioms.

6.1. NLP axioms

Of the various NLP axioms we have developed, we show two instances. Table 1 lists axioms dealing with possessives, whereas Table 2 handles sentences that use the coordinative conjunction *'dan'*.

Using the axioms listed in Table 1, the two phrases presented in Section 5.4 above yield the same canonical logical form. In fact, all paraphrases signifying possession will entail the canonical logical form. The axioms are similar to production rules: if the combination of facts on the left-hand side are found to appear in the KB, the axiom will assert the right-hand side literals as new facts. For example, the first possessive axiom states that if the literal *possess(A,X)* is found, then *event(E,memiliki)*, *agent(E,A)*, and *patient(E,X)* will also be asserted, with the corresponding variables bound to the concepts specified in the KB.

Table 1: Axioms handling possessives

{possess(a,x)}	→	*{event(e,memiliki)* ∧ *agent(e,a)* ∧ *patient(e,x)}*
{object(x,CONCEPT1) ∧ *nobject(m,milik)* ∧ *person(a,CONCEPT2)* ∧ *nn(m,a)* ∧ *nn(x,m)}*	→	*{event(e,memiliki)* ∧ *agent(e,a)* ∧ *patient(e,x)}*
{person(a,pemilik) ∧ *nn(a,x)}*	→	*{event(e,memiliki)* ∧ *agent(e,a)* ∧ *patient(e,x)}*

Table 2 shows the axiom that handles sentences using the coordinative conjunction *'dan'*. Previously, for any coordination that holds between concepts d_1 and d_2, the syntax-driven analysis simply introduces a new concept d representing the conjunction of the two concepts. The conjunction axiom searches for all literals in which d participates as an argument, and asserts new copies of those literals in which d_1 and d_2 appear in place of d.

Table 2: Axiom handling coordinative conjunction

$\{dan(d,d_1,d_2) \wedge PRED_1(\ldots,d,\ldots) \wedge \ldots \wedge PRED_n(\ldots,d,\ldots)\}$	→	$\{PRED_1(\ldots,d_1,\ldots) \wedge PRED_1(\ldots,d_2,\ldots) \wedge \ldots \wedge PRED_n(\ldots,d_1,\ldots) \wedge PRED_n(\ldots,d_2,\ldots)\}$

6.2. World knowledge axioms

In our work, we also provide additional information to the system that is derived from a prototype Indonesian WordNet in the form of world knowledge axioms[2]. These axioms analyse the semantic representations constructed through syntax-driven analysis and will add literals that improve the inferential capabilities of the system. There are four types of world knowledge axioms: synonym axioms, antonym axioms, hypernym axioms, and derivational morphology axioms.

WordNet (Fellbaum, 1998) is a lexical resource where specific senses of words are clustered together into synonym sets, and semantic relationships between these sets are specified. (Putra et al., 2008) presents work on the development of an initial Indonesian WordNet[3]. For our purposes, this Indonesian WordNet can be viewed as a collection of Prolog facts stating semantic relationships holding between intrinsic symbols denoting domain concepts, e.g.

synonym(ibu,bunda).
antonym(panas,dingin).
hypernym(kue,makanan).

The **synonym** axiom (Table 3) is designed for nouns, verbs, adjectives, and adverbs. For each domain concept appearing in the semantic representation of declarative sentences, it will assert new literals based on the Indonesian WordNet.

Table 3: Axiom handling synonyms

$\{synonym(k,k_1) \wedge \ldots \wedge synonym(k,k_n) \wedge PRED(\ldots,k,\ldots)$	$\rightarrow$	$\{PRED(\ldots,k_1,\ldots) \wedge \ldots \wedge PRED(\ldots,k_n,\ldots)$

The **antonym** axioms (Table 4) assert new literals explicitly stating the negation of the opposing concept of adjectives appearing in the semantic representation.

Table 4: Axioms handling antonyms

$\{antonym(VAR1,VAR2) \wedge not(a,b) \wedge property(a,VAR1) \wedge isAdjNGrade(VAR1)\}$	$\rightarrow$	$\{property(b,VAR2)\}$
$\{antonym(VAR1,VAR2) \wedge not(a,b) \wedge property(a,VAR2) \wedge isAdjNGrade(VAR2)\}$	$\rightarrow$	$\{property(b,VAR1)\}$
$\{antonym(VAR1,VAR2) \wedge property(a,VAR1)\}$	$\rightarrow$	$\{not(a,b) \wedge property(b,VAR2)\}$
$\{antonym(VAR1,VAR2) \wedge property(a,VAR2)\}$	$\rightarrow$	$\{not(a,b) \wedge property(b,VAR1)\}$

The **hypernym** axiom (Table 5) is designed for nouns and verbs. It adds all new hypernym literals of all the concepts in the semantic representation.

Table 5: Axiom handling hypernyms

$\{hypernym(k,k_1) \wedge \ldots \wedge hypernym(k,k_n) \wedge PRED_{def}(VAR,k) \wedge PRED_1(\ldots,VAR,\ldots) \wedge \ldots \wedge PRED_m(\ldots,VAR,\ldots)\}$	$\rightarrow$	$\{PRED_{def}(VAR_1,k_1) \wedge \ldots \wedge PRED_{def}(VAR_n,k_n) \wedge isA(VAR_1,VAR) \wedge \ldots \wedge isA(VAR_n,VAR) \wedge PRED_1(\ldots,VAR_1,\ldots) \wedge \ldots \wedge PRED_1(\ldots,VAR_n,\ldots) \wedge \ldots \wedge PRED_m(\ldots,VAR_1,\ldots) \wedge \ldots \wedge PRED_m(\ldots,VAR_n,\ldots)\}$

The **derivational morphology** axioms (Table 6) can be seen as introducing frame-theoretic knowledge to the QA system. Specifically, it establishes a logical link between the semantic representations of intransitive and transitive verbs. Using this axiom, sentences containing

[2] Note that some would take issue with our use of the term '*world knowledge*', as WordNet is, strictly speaking, a lexical semantics resource, unlike, say, OpenCyc.
[3] http://bahasa.cs.ui.ac.id/iwn

verbal phrases that consist of an intransitive verb and obligatory complement noun phrase (*pelengkap*) will have the same semantic representation as a sentence with an active transitive verb. These axioms also handle derivational morphosemantic relations. They are designed to equate the semantic representations of noun phrases signifying *profession* and sentences containing verbs signifying *profession*. Specifically, the presence of *profession(X,Y)* intrinsic noun literal results in the assertion of appropriate *agent* and *event* literals.

Table 6: Axioms handling derivational morphology

{der(VAR1,VAR2) ∧ vintr(VAR1) ∧ vtran(VAR2) ∧ event(e1,VAR1) ∧ theme(e1,t) ∧ object(t,VAR3)}	→	*{event(e2,VAR2) ∧ patient(e2,t)}*
{der(VAR1,VAR2) ∧ person(VAR1) ∧ vtran(VAR2) ∧ profession(a,VAR2) ∧ nn(a,x)}	→	*{event(e,VAR2) ∧ agent(e,a) ∧ patient(e,x)}*
{der(VAR1,VAR2) ∧ person(VAR1) ∧ vintr(VAR2) ∧ profession(a,VAR2) ∧ nn(a,x)}	→	*{event(e,VAR2) ∧ agent(e,a) ∧ theme(e,x)}*
{der(VAR1,VAR2) ∧ person(VAR1) ∧ vintr(VAR2) ∧ profession(a,VAR2)}	→	*{event(e,VAR2) ∧ agent(e,a)}*

7. Axioms in Action

Our QA system is implemented in Prolog. It consists of a DCG grammar, where each syntactic rule has been augmented with semantic attachment rules (see Section 5.3), and we also constructed a small handcrafted lexicon where words were associated with lexical semantics as discussed in Section 5.2. A Prolog parser with semantic representation building, using the associated attachment rules, was developed to handle our resources, and testing revealed that indeed the correct semantic representations were being transduced from input sentences in Indonesian.

The next step was to develop a question answering module that employs the axioms described in Section 6. In general, semantic representations of declarative sentences are asserted as new facts to the KB, but not before passing them through the axiom post-processing. This can be repeated for as many sentences as necessary. Finally, we issue a query to the knowledge base by asking it an Indonesian interrogative sentence.

We first show how the system can still answer simple questions without the need for axioms as defined in Section 6 above. Consider the following sentence:

Lusi mencicipi kue buatan ibu dan apel hijau yang dibeli kakak di toko Harun
Lusi taste cake made by mother and apple green that bought sister at store Harun
"Lusi tastes a cake made by mother and a green apple that sister bought from Harun's store."

Without the aid of axioms, the semantic analyzer constructs the following semantic representation:

```
event(e,mencicipi), agent(e,a), patient(e,p), person(a,lusi), dan(p,x,y),
object(x,kue), nobject(n,buatan), ibu(i,ibu), nn(x,n), nn(n,i),
object(y,apel), property(w,hijau), attrib(y,w), event(f,membeli),
agent(f,k), patient(f,y), person(k,kakak), di(f,t), object(t,toko),
person(h,harun), possess(h,harun)
```

This representation can answer simple questions such as:

Siapa membeli apel ?
Who buy apple ?
"Who bought an apple?"

which yields the correct answer '**kakak**' (sister), since it unifies with the following query:

```
ans(Ans) :- person(X1,Ans),event(X2,membeli),agent(X2,X1),patient(X2,X3),
            object(X3,apel)
```

However, when asked a different question such as the following one:

Lusi mencicipi apa ?
Lusi taste what ?
"What did Lusi taste?"

which produces the following query:

```
ans(Ans) :- person(X1,lusi),event(X2,mencicipi),agent(X2,X1),
            patient(X2,X3), object(X3,Ans)
```

the system is unable to identify the correct answer, since the patient of the "**mencicipi**" event is **p**, a domain concept introduced to represent the conjunction (see the **dan(p,w,x)** literal). By employing the coordinative conjunction axiom, the semantic representation will be augmented with the literals **patient(e,x)** and **patient(e,y)**, thus making the representation canonical. As a result, the system unifies the query with '**kue**' (cake) and '**apel**' (apple) for the answer.

The final example shows the value of the world knowledge axioms. Firstly, due to the existence of a WordNet fact **synonym(mencicipi,memakan)**, the synonym axiom asserts new literals for the synonym of 'mencicipi' (taste) term that is 'memakan' (eat). Specifically, it asserts **event(e2,memakan)**, **agent(e2,a)**, and **patient(e2,p)**. On the other hand, due to the existence of an Indonesian WordNet fact **hypernym(apel,buah)**, the hypernym asserts new literals for the hypernym of 'apel' (apple) term that is buah (fruit). Specifically, it asserts **object(q,buah)**, **isA(y,q)**, **dan(p,x,q)**, and **patient(e,q)**. As a result, given the following question:

Buah	apa	yang	dimakan	Lusi	?
fruit	what	that	eaten by	Lusi	?

"What fruit did Lusi eat?"

which produces the following query:

```
ans(Ans) :- event(X1,memakan),agent(X1,X2),patient(X1,X3),person(X2,lusi),
            object(X3,Ans), object(X4, buah), isA(X3,X4).
```

the system will produce the correct answer '**apel**' (apple), whereas without the world knowledge axioms it fails to do so.

8. Discussion and Summary

The implemented axioms have been shown to increase the capability of our Indonesian QA system in answering questions with syntactic variations and use of implicit world knowledge. We believe that handling these aspects as logical axioms is the right strategy, as it is at the appropriate level of abstraction, and lightens the burden on designing the semantic attachment rules that are still fairly tightly coupled to syntactic structure. Moreover, these axioms are not necessarily specific to the Indonesian language. For instance, the world knowledge axioms are fairly language independent, although the NLP axioms may have to be revised for another language, depending on how certain concepts are conveyed, e.g. possessives. Our prototype Prolog system still employs a fairly simple inference mechanism. In the future, we hope to feed the semantic representations into more sophisticated theorem provers, similar to the approach in (Blackburn and Bos, 2005).

References

Alwi, H., S. Dardjowidjojo, H. Lapoliwa and A. Moeliono. 1998. *Tata Bahasa Baku Bahasa Indonesia*. Balai Pustaka.

Blackburn, P. and J. Bos. 2005. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. CSLI Publications.

van Durme, B., Y. Huang, A. Kupść and E. Nyberg. 2003. Towards Light Semantic Processing for Question Answering. *Proceedings of the 2003 Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Text Meaning*, pp. 54-61.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Hobbs, J. 1985. Ontological Promiscuity. *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 61-69.

Joice. 2002. *Pengembangan lanjut Pengurai Struktur Kalimat Bahasa Indonesia yang menggunakan Constraint-Based Formalism*. Undergraduate thesis, Faculty of Computer

Science, University of Indonesia.

Jurafsky, D.S. and J.H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall.

Larasati, S.D. and R. Manurung. 2007. Towards a Semantic Analysis of Bahasa Indonesia for Question Answering. *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING 2007)*.

Moldovan, D., C. Clarke, S. Harabagiu and S. Maiorano. 2003. COGEX: A Logic Prover for Question Answering. *Proceedings of the 2003 Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 87-93.

Narayanan, S. and S. Harabagiu. 2004. Question Answering based on Semantic Structures. *Proceedings of the 20th International Conference on Computational Linguistics*.

Purwarianti, A., M. Tsuchiya and S. Nakagawa. 2007. A Machine Learning Approach for Indonesian Question Answering System. *Proceedings of the International Conference on Artificial Intelligence and Applications (AIA 2007).*

Putra, D.D., A. Arfan and R. Manurung. 2008. Building an Indonesian WordNet. *Proceedings of the 2nd International MALINDO Workshop*.

Shen, D. and Lapata, M. 2007. Using Semantic Roles to Improve Question Answering. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 12-21.

Wijono, S.H., I. Budi, L. Fitria and M. Adriani. 2006. Finding Answers to Indonesian Questions from English Documents. *Working Notes of the Workshop in Cross-Language Evaluation Forum (CLEF 2006)*.