UNDERSTANDING AND EXECUTING A DECLARATIVE SENTENCE INVOLVING A FORMS-OF-BE VERB

W. Faris[1] and K. Cheng

Department of Computer Science
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

UH-CS-09-03

February 18, 2009

Keywords: Semantics, Grammar, Natural Language Processing, Artificial Intelligence.

Abstract

An important criterion for any artificial intelligence program is its capability to communicate with the external world. One common form of communication is by the use of a natural language, specifically English. When provided a sentence, it is important for the program to understand the intention of the given sentence which is an important first step for a program to perform logical reasoning. In this paper, we discuss two components of the grammar that affects the understanding of a sentence: role and control. These two components represent the knowledge that teaches how to use the language to express a thought. We describe in detail what needs to be learned for each of these components for three major grammar terms: noun phrase, declarative sentence, and forms-of-be verb. We then show how to use them to create a declarative thought corresponding to a given declarative sentence that uses a forms-of-be verb. Finally, we show what needs to be learned by the program so that the declarative thought can be understood correctly based on exactly what is involved in the subject and the predicate of the given sentence. Object-Oriented paradigm is used to analyze the problem and design the solution to attack the problem.

[1]W. Faris is supported in part by GAANN: Doctoral Training in Computer and Computational Sciences, US Department of Education under grant #P200A070377.

UNDERSTANDING AND EXECUTING A DECLARATIVE SENTENCE INVOLVING A FORMS-OF-BE VERB

W. Faris[1] and K. Cheng

Abstract

An important criterion for any artificial intelligence program is its capability to communicate with the external world. One common form of communication is by the use of a natural language, specifically English. When provided a sentence, it is important for the program to understand the intention of the given sentence which is an important first step for a program to perform logical reasoning. In this paper, we discuss two components of the grammar that affects the understanding of a sentence: role and control. These two components represent the knowledge that teaches how to use the language to express a thought. We describe in detail what needs to be learned for each of these components for three major grammar terms: noun phrase, declarative sentence, and forms-of-be verb. We then show how to use them to create a declarative thought corresponding to a given declarative sentence that uses a forms-of-be verb. Finally, we show what needs to be learned by the program so that the declarative thought can be understood correctly based on exactly what is involved in the subject and the predicate of the given sentence. Object-Oriented paradigm is used to analyze the problem and design the solution to attack the problem.

Index Terms

Semantics, Grammar, Natural Language Processing, Artificial Intelligence

I. INTRODUCTION

There are several major criteria for any artificial intelligence program. One of them is its ability to communicate with the external world using a natural language, specifically English. In order to communicate successfully, one requirement is the program's ability to comprehend the intention of the given sentence. We divide the problem into three sub-problems. The first sub-problem is to learn the grammar of the English language and how to use it to express thoughts. The second sub-problem involves parsing an input English sentence using the learned grammar to produce an internal thought. The last sub-problem is to understand the intention of the internal thought produced by the parsing process.

There is a substantial amount of research being done in the field of natural language processing. Some of the major challenges and barriers facing research in the field are presented in [1]. Most research tries to overcome these challenges using logic based or functional programming languages. Several books, such as [2], [3], and [4], have been written that describe how to use the logic-based programming language Prolog to create natural language applications, such as the CHAT-80 system [5]. Functional programming languages, such as Miranda [6], Haskell [7], and OCaml [8] have been used to develop several functional language parsers [9-11]. Using these parsers, many natural language interfaces such as Lolita [12] and Windsor [13] have been built. A detailed survey is provided in [14] on natural language interfaces that use lazy functional programming. Along with logic-based languages, for every verb introduced using functional programs, a new function for that verb needs to be written into the system. Carrying out the intention of the verb may be implicit to the execution of the functions. However,

[1]W. Faris is supported in part by GAANN: Doctoral Training in Computer and Computational Sciences, US Department of Education under grant #P200A070377.

it is unclear how to locate the starting point of the reasoning process where the understanding of the sentence occurs. Another area of natural language research is from the linguistics point of view, focusing on the different formal models of grammar. The most common model is the context free grammar model, popularized by Chomsky [15] and later Backus and Naur [16], and has lead to many variations, such as Restriction Language [17], attributed [18], Head-Driven Phrase Structure [19], construction [20] and Tree Adjoining [21]. Besides the context free model, grammars such as dependency [22], augmented transition network [23], categorical [24], Montague [25], combinatorial categorical [26], semantic [27], and type-theoretic [28] have been investigated. A major problem on using a formal grammar model is that additions to accommodate a new aspect of the natural language may require a major change to many existing structures [29].

Some researchers are interested in the production of parts-of-speech tags when given a sentence [30, 31]. Recently, we have developed a sub-system to learn the grammar terms (parts-of-speech) of the English language [32]. It is a part of the communication agent of the learning program in the project: A Learning Program System (ALPS) [33] whose goal is to learn all knowledge. The initial focus of ALPS has been on the development of the memory agent of a multi-agent artificial intelligence [34] program to store knowledge and the relationships among them. Basic capabilities, such as creating a new category, adding objects, attributes, and properties to a category have been provided. We have developed two major knowledge components of categories: hierarchy [35] and definition [36]. Hierarchy may be used to specify both generalization and containment relationships among knowledge. Definition specifies the necessary and sufficient condition that is used to classify objects as a specific category. The sub-system in [32] first learns a subset of the English grammar, and then uses the grammar to parse sentences. A key idea introduced is the role of a grammar term which defines the intention of the term. The roles of the various grammar terms in a particular sentence allow the program to understand the exact purpose of the sentence. They serve as the bridge between the grammar knowledge world and the knowledge world that the sentence is trying to express. For our parsing algorithm, an appropriate role has been correctly identified for every part of the sentence. However, in order to fully understand the details of the sentence so as to carry out its intention, the detailed content of each individual role needs to be organized in a meaningful way. We call this organization effort by a role a satisfaction of the role. For those grammar terms that have multiple satisfied sub-roles, there are two tasks that need to be accomplished. The first task is to decide which role should be used to carry out this satisfaction process, and we introduce the idea of control to accomplish this. The second task is to figure out what knowledge is needed in order to correctly satisfy the content of the role. We describe in this paper the necessary knowledge needed for the roles of three major grammar terms: noun phrase, declarative sentence, and the forms-of-be linking verb. In addition, we are focusing on the understanding of a declarative sentence that uses a forms-of-be linking verb. The knowledge required for the understanding of a declarative sentence that uses an action verb can be found in [37].

The rest of the paper is organized as follows. Section 2 gives a short description of the English grammar learned in [32] and briefly describes our parsing algorithm. Section 3 describes in details our solution to satisfy the role for the noun, the noun phrase, and the forms-of-be linking verb. Section 4 looks at how to generate the correct thought given a declarative sentence that uses a forms-of-be verb. Section 5 discusses how this thought is then understood by the system based on the different definitions that need to be learned. Section 6 concludes our paper with a discussion and a look at future work.

II. The English Grammar

The learning of the English grammar is done in an incremental manner in ALPS. Our program first learns a subset of the English grammar, and then uses it to parse and understand English sentences. In the future, it will also use the learned grammar to generate an English sentence when given a thought. Our program first learns the various grammar terms in English: such as sentence, complete subject, verb, noun phrase, and preposition. There are four major components introduced in [32] for each grammar term: structure, role, kind, and rule. Not all components are required for every grammar term, which means that a specific grammar term may be defined by a combination of some of these components. The structure of a grammar term may be either a sequence or an alternative, which defines exactly the grammatical format of the term and the type of knowledge that it contains. The role of a grammar term defines the intention of the term. They serve as the bridge between the grammar knowledge world and the knowledge world that a sentence is trying to express. A term may have multiple kinds, which are subsets that may share the same structure but must have different roles. For example, a declarative sentence is one kind of the sentence grammar term. Finally, a rule specifies a condition that must be satisfied. Rules may be applied directly to the grammar term or to one of its structures. The control is a recently added fifth component that helps define a grammar term, and it will be discussed more throughout this paper. Figure 1 shows an example of how the grammar term, sentence, is defined by its components.

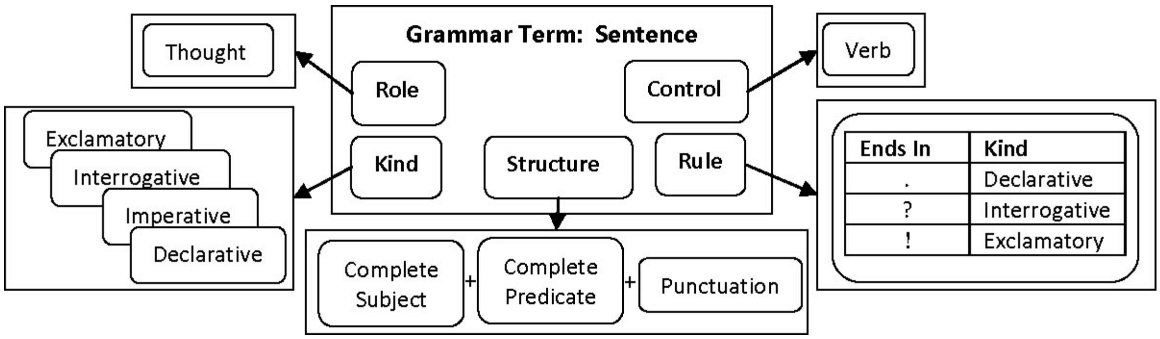


Fig. 1. The components of the grammar term: sentence.

Our parsing algorithm consists of a syntactic stage followed by a semantic stage. The syntactic stage deals with the analysis of the individual words in the sentence. The syntactic stage in [32] simply stems words into its root form and passes the re-constructed sentence to the semantic stage. Substantial extensions such as morphology and multiple word terms have just been added in the syntactic stage [38]. The semantic stage deals with recognizing all the subparts of a given sentence, identifying the knowledge referenced in that sentence and producing an appropriate thought associated with the sentence. The semantic stage is based on a high-level template parser that makes use of the individual structures' unique internal parsers. It is a depth-first top-down parser whose execution consists of processing in two major manners: top-down and bottom-up. The top-down processing starts parsing using the structure of the highest grammar term: the sentence, and works its way down to lower level terms. It is responsible for recognizing the terms of the sentence, and identifying the knowledge mentioned in the sentence. After recognizing a term of the sentence by the knowledge involved, the algorithm verifies that the appropriate rules unique to the grammar term or its structure are satisfied. Based on the recognized grammar term with all its identified sub-roles, the top-down parsing process finishes by choosing the associated role of the term and including that role in the parse result. The role is the high-level purpose of the grammar term. However, the detailed content of the role needs to be organized in order to understand the purpose of the term. The processing in the bottom-up manner satisfies the role by organizing all the identified sub-roles appropriately, and the resulting role is returned to a higher-level term.

There are two possible cases in satisfying a role. First, when parsing a grammar term that has an alternative structure, since the parsing is completed by only one alternative, the grammar term can have at most one satisfied sub-role. If the alternative is at the lowest level, the role is satisfied by the knowledge identified in the sentence, e.g., the knowledge representing John Smith or buy. Otherwise, this role can easily be satisfied by the knowledge stored in the role of the involved alternative. On the other hand, after parsing a grammar term that has a sequence structure, the role of each term in the sequence has been satisfied, so there are several sub-roles available to satisfy the role associated with the grammar term. There are two tasks involved. The first task is to identify the correct sub-role to carry out the satisfaction process. We introduce the idea of control to identify the correct sub-role to carry out the duty of satisfying a role. The second task is on how to carry out the process once the correct role has been identified. It has to prepare its content in such a way that the intention can be understood and executed easily by the appropriate knowledge object. By the time the sentence grammar term finishes parsing, the result is a thought that reflects what the sentence wants to express, and the identified knowledge is stored as the appropriate parts of this thought.

We will present the details on how the control chooses the proper role, how the chosen role carries out the satisfaction process, and present what needs to be learned to accomplish these tasks. Specifically, we describe in detail the roles for noun phrase, declarative sentence, and the forms-of-be linking verb. The noun phrase is associated with the usage role while its meaning depends on where it is used. The declarative sentence has the role of declaration, meaning it declares a statement about a knowledge object. One form of a declarative sentence uses the forms-of-be linking verb: 'is', 'am', and 'are'. The forms-of-be verb is associated with the define role which is used to define a knowledge object as a state of being, an aspect of a knowledge object, or a relationship. We will describe in detail how the define role understands the intention of the sentence. Another form of a declarative sentence makes use of an action verb. The details on how act-role, the role for action verbs, prepares and understands a given sentence can be found in [37].

III. Satisfying the Usage Roles

The satisfaction of the usage role is the most complicated case in our program. The usage role serves a different purpose depending on its position in a given sentence. It is used by the grammar terms noun and noun phrase. To understand how the usage role is used, it is important to first understand how the ALPS program learns about the noun phrase grammar term. A noun phrase consists of a sequence of grammar terms and may contain multiple nouns. It starts with an optional determinate, followed by a compulsory noun, sometimes known as the head noun, and finally a number of optional prepositional phrases. Each prepositional phrase consists of a preposition followed by a noun. Thus, the following are all acceptable noun phrases: "John", "a dog", and "the king of Spain". The compulsory noun is given the usage role with the name simple subject. The noun in any optional prepositional phrase is also a usage role but referred to as the object of the preposition. The exact usage of the simple subject, the object of the preposition, and their relationships are different for different structures of the usage role.

The structure of a usage role helps to reflect the knowledge represented by the noun or noun phrase. We classify five different kinds of structures each implemented by its own class. They are whole-object, constant, aspect-of-an-object, relationship, and additional-information. The whole-object structure is used for the following two situations. The first situation is for proper nouns represented by simple knowledge objects, such as "John Smith", "USA" and "the Pacific Ocean". The second situation involves an optional determinate followed by a category name, e.g., "3 apples", "a house", and "2 cars". The grammar term determinate includes the alternatives article and cardinal number. An article can be 'a', 'an', or 'the', and the value for all of them is one. A cardinal number is currently learned to be recognized by the natural number data type. As a result, it can only recognize actual numerical numbers. To recognize and understand cardinals such as one, five, twenty, and thirty, the program needs to learn the spelling and rules for numbers. The second structure of usage role is the constant structure. It is used for storing a simple numeric constant, which does not have a subsequent category name, such as "2". It is also used for concept values such as "3 meters", "10 grams", and "4 yards 1 foot 2 inches". These two structures are used for noun and any noun phrase that does not contain a prepositional phrase.

The next two structures of the usage role (aspect-of-an-object and relationship) deal with a relationship between knowledge objects. A common way the English language to express the knowledge that corresponds to an aspect of an object is by using a noun phrase where the simple subject is followed by the preposition 'of' and then by the object of the preposition: such as the queen of England, the weight of John, and the average of the radii of the circles. Currently, only the preposition 'of' has been taught to have the aspect-of-an-object structure. Every instance of this structure has two logical knowledge objects: *aspect* and *object*. In order to bridge an aspect-of-an-object structure with the grammar, the system needs to learn the link between these two logical objects and the usage roles of a noun phrase. This means teaching this role structure to associate *aspect* with the simple subject and *object* with the object of the preposition. By using this learned knowledge, the role structure can be satisfied correctly, and the complete meaning of the noun phrase can be recognized. The next structure of a usage role is the relationship structure. It is used for situations such as "the boy across the street", "the book under the table", and noun phrases that use other location prepositions. The two logical knowledge objects of the relationship structure are the *object-of-interest* and the *reference-object*. The *object-of-interest* is learned to associate with the simple subject and the *reference-object* with the object of the preposition.

Finally, the last structure of a usage role is the additional-information. It is used to provide the sentence with supplementary information. Many prepositions are currently learned to have this structure, e.g., 'to', 'from' and 'for'. Besides the preposition used, the other important local logical knowledge object is the *purpose*. The *purpose* is learned to associate with the object of the preposition. As in other noun phrases that involve prepositional phrases, there may be a sequence of additional-information prepositional phrases, each independent of one another such as in "John buys 8 apples from Mary for 2 dollars."

Now, we will describe how the two components of a usage role, its purpose and structure, are satisfied for a noun in a noun phrase and a noun phrase in a sentence. First, we describe how the usage role of a noun is satisfied. Its name represents its purpose and is determined by its position in a noun phrase, namely simple subject and object of the preposition. Its structure is either a whole-object or a constant. A whole-object corresponds to knowledge whose name has been learned, and thus can be searched successfully when parsed. On the other hand, data type or concepts recognize constants and create them when parsed.

Next, we describe how the structure of the usage role is satisfied for a noun phrase. At the time of satisfying the role for a noun phrase, several satisfied sub-roles may be available. In order to recognize which structure the given noun phrase should have, the noun phrase uses the control knowledge to locate the correct role structure. For the noun phrase, the control knowledge is given by the sequence: preposition, noun. This means that if there is a prepositional phrase, the correct structure will be dictated by the preposition used in the sentence; otherwise it is dictated by the compulsory noun in the noun phrase. If the structure is dictated by the noun, then it has already been satisfied correctly by the noun at a lower level. If the structure is dictated by a preposition, which has learned to have a specific structure, an object of that structure is created with its content organized accordingly. For example, an aspect-of-an-object structure is created if the preposition is 'of'.

We have just described how a noun phrase is structured. The following describes what purpose the noun phrase serves. Similar to the nouns in the noun phrase, the noun phrase itself serves a different purpose depending on its position in a given sentence. In a simple declarative sentence, the noun phrase that appears before the verb is known as the complete subject and serves the role of subject. Here subject means the object of interest of the sentence. For noun phrases that appear after the verb, its purpose depends on the type of verb used. If the verb is a linking verb, then the noun phrase that follows is the subject complement grammar term. The purpose of this noun phrase is to serve as the predicate of the sentence. If the verb is an action verb, then the noun phrase that follows is a part of the object complement grammar term. This noun phrase serves the purpose of the direct object of the sentence. Within the object complement, an optional noun phrase may precede the direct object, which serves as the indirect object.

To show how a noun phrase is satisfied, suppose the noun phrase "the weight of John" is the complete subject of a given declarative sentence. As the phrase is being parsed, the term 'weight' is recognized by the concept knowledge. This will create a whole-object structure, which is satisfied by storing the concept knowledge weight as the simple subject of the noun phrase. Following this, the next term parsed is the preposition 'of'. Since this preposition is taught to be a usage role with an aspect-of-an-object structure, such a structure is created. However, this structure cannot be satisfied yet because not all the necessary component knowledge has been parsed. Next, 'John' is recognized as an object of a category. Another usage role with whole-object structure is created and satisfied, but this usage role is stored as the object of the preposition of the noun phrase. Now that each grammar term in the noun phrase sequence has been parsed, the top-down stage of the parsing of the noun phrase is completed and the role for the noun phrase will be satisfied. Since the noun phrase contains multiple roles, the control is used to determine which role and its structure will be used to satisfy the usage role of the entire noun phrase. For the noun phrase, the first control object is the preposition. Therefore, an aspect-of-an-object structure is created in this example. It is now possible to satisfy this structure, since all the required components are available. Satisfying this structure involves using the bridge knowledge that matches the grammar knowledge of the usage role with the logical knowledge for the structure. For the aspect-of-an-object structure, the bridge links simple subject to *aspect* and object of the preposition to *object*. Using this bridge, the knowledge identified as the simple subject and the object of the preposition are stored in the structure as the *aspect* and *object*, respectively. In addition to satisfying the structure of the noun phrase, its purpose also needs to be identified. Since this noun phrase is the complete subject of the sentence, it assumes the role of the subject of the sentence. The process of satisfying usage roles for other structures is accomplished in a similar manner.

IV. The Declaration and Define Roles

The purpose of a sentence is to express different kinds of thoughts: to make a declaration, to ask a question, or to issue a command. Therefore, the role of a sentence is taught to be a thought. There are several different kinds of thought roles, each corresponding to a different kind of sentence. A declarative sentence is taught to have the declaration role, an interrogative sentence has the question role, and an imperative sentence has the command role. To respond to a thought after understanding its purpose, the program may carry out the intent of a declaration, respond to the query of a question, or execute the order of a command.

When the system is given an English sentence, the parsing algorithm will use the learned grammar, beginning with the sentence grammar term, to parse the sentence. For each successfully parsed grammar term or structure, its associated role will be identified and called to satisfy the detail contents. For example, all relevant knowledge involving the complete subject is satisfied by the subject role. The satisfaction of the roles is accomplished in a bottom-up manner. By the time all the components of the sequence structure for a sentence have been completed, all the sub-roles have been satisfied individually and all that remains is the role at the highest level, the role of the sentence. Now, suppose a given sentence ends with the punctuation '?'. The rule attached to the sentence will decide that it is an interrogative sentence with the question thought. If our given sentence ends with the punctuation '.', then the rule determines that it is a declarative sentence. All the roles that have been collected are stored in a language declaration thought. This thought maps the roles with the corresponding knowledge found in the sentence.

At this point, all the roles in the language declaration thought have been satisfied except the declaration role associated with the entire sentence. Since there are several satisfied sub-roles such as the subject, the verb, and the predicate roles, the control tells our program which role to carry out the satisfaction process of a declaration thought. The control for a declarative sentence is taught to be the verb. Now suppose the verb in the sentence is an action verb, the identified and chosen verb role is the act role. The knowledge learned by the act role, and how act role uses these learned knowledge to recognize all the potential effects of a sentence can be found in [37]. On the other hand, if the English sentence uses a forms-of-be verb, the chosen verb role to satisfy the declaration thought is the define role. Satisfying the define role is similar to the process of satisfying usage roles. The define role has two local logical knowledge objects: the *object-of-interest* and the *definition*. Its purpose is to use the *definition* to define the *object-of-interest*. In English, the subject represents the *object-of-interest*, and the *definition* is provided by the predicate. The relationships learned in the define role are displayed in Figure 2. Using this learned knowledge will allow the define role to be satisfied correctly to create the final declaration thought. The final declaration thought uses the knowledge from the define role and the language declaration thought to link the local logical knowledge objects with the knowledge found in the sentence. Figure 3 shows the two declaration thoughts for the given sentence: "Mary is a teacher." Figure 3(a) shows the language declaration thought indexed by language knowledge terms, while Figure 3(b) shows the final declaration thought where the indexing have already been changed to the logical knowledge of the define role. The completely satisfied final declaration thought is the result of the parsing algorithm. Figure 4 outlines the process of parsing a declarative sentence to produce a final declaration thought.

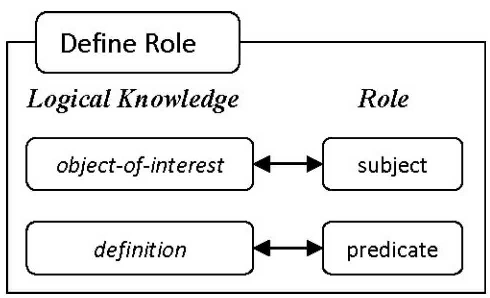
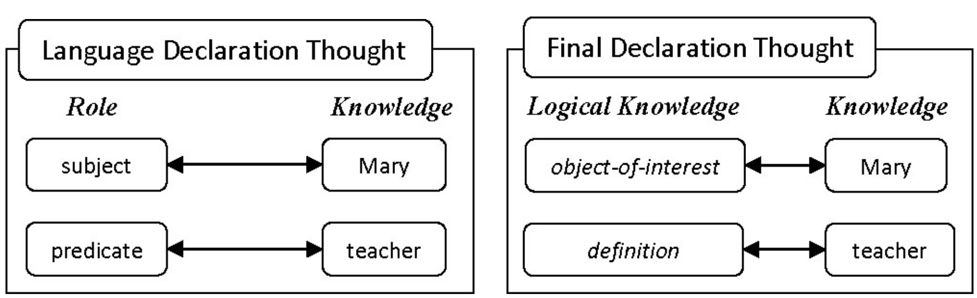


Fig. 2. The relationships learned by the define role.



(a) Language Declaration Thought (b) Final Declaration Thought

Fig. 3. The declaration thoughts for the sentence "Mary is a teacher."

Fig. 4. The process of parsing a sentence to produce a final thought.

V. Understanding and Executing a Sentence Using a Forms-of-be Verb

The final sub-problem in comprehending a declarative sentence using a forms-of-be verb as the main verb is to understand the final declaration thought produced by the parsing algorithm as described in the previous section. Grammatically, the forms-of-be verb is used to link the predicate to the subject. Semantically, the system already knows that the intention of the given declarative sentence is to use the *definition* to define the *object-of-interest*. However, a definition here is very ambiguous. For instance, the following two sentences both link the predicate to the subject at the grammatical level: "Gorillas are primates." and "Steve is a doctor." Yet, on the semantic level, the first sentence defines the category of gorillas as a sub-category of primates, while the proper noun Steve is defined as having the occupation of a doctor in the second sentence. So how can two grammatically identical sentences be understood differently? We solve this problem by recognizing that there are many kinds of definitions. For example, one kind of definition is to define an unknown as a kind of knowledge, e.g., "Country is a category." So if the *object-of-interest* is unknown to the system, and the *definition* is a recognized kind of knowledge, then the declarative sentence is to define the unknown as that kind of knowledge. Similarly, another kind of definition is to convey the idea that one category is a generalization of another category, e.g., "Apples are fruits." In such a definition, the general category is the *definition*, and the specialized category is the *object-of-interest*. So when we recognize both *object-of-interest* and *definition* are categories, we can infer that the declarative sentence is to define a generalization relationship between the two categories.

As a result, we use a pattern to distinguish the different kinds of definitions. A pattern consists of two classifications, which reflect the *object-of-interest* and *definition*, respectively. For each pattern, the meaning and appropriate action is provided to inform the system of the kind of definition this particular pattern represents. By finding the classification of the knowledge stored in the final declaration thought and using the knowledge the recognized pattern implies, the intention of a given declarative sentence can be understood. In ALPS, there are currently six types of classifications: kind, category, object, concept, concept value, and unknown. The kind classification represents internal bodies of knowledge that are recognized by the system, this includes category, object, idea, concept, etc. This classification is used to create a new instance of one of these bodies of knowledge. Concept represents the bodies of quantifiable knowledge that has units of measurement, such as height, weight, speed, and volume. A concept value is a constant value of a concept such as 7 grams. Finally, unknowns classify those terms that are not recognized by ALPS where each term represent a new piece of knowledge.

Based on the implications of the learned patterns, the actual intention of the given declarative sentence can now be understood. Take the first example sentence that involves gorillas. Assume that both gorilla and primate have been learned as categories by the ALPS system, our system recognizes it as the pattern <category, category>. The action attached to this pattern creates a hierarchy relation with gorilla as a child of primate [35], which is determined to be the semantic meaning of the sentence. As for the other sentence "Steve is a doctor", assume that the knowledge base of the ALPS system contains Steve as an object of the human category, and doctor as a category taught as a sub-category of occupation. Our system will recognize this as the pattern <object, category>. For this pattern, the action associated with it is to create an attribute for the object that has category as its aspect value. In other words, doctor becomes an attribute value of the object Steve. However, an internal condition needs to be verified before this relationship can be formed. In order for Steve to have an attribute of doctor, the system needs to verify that the category human can have occupation as an aspect. The establishment of this relationship is discussed shortly when we present the possess role. Once it asserts that humans can have an occupation, the definition that the occupation of Steve is a doctor is logically possible. The verification of this condition ensures that Steve the human can be a doctor, but that Steve the cat cannot.

As mentioned earlier, we need to teach the system that humans can have an occupation. One alternative is to use the existing special interface of the learning program. Alternatively, this may be achieved conveniently by teaching the system the declarative sentence "Humans have occupations." Similar to the define role and its relation to the forms-of-be verb, the forms-of-have verb is taught to have the possess role. Given a sentence, the purpose of the possess role is to attach the predicate to the subject. In our example, the subject human (singular after stemming) can possess an occupation. Again, in order to distinguish the wide range of semantic meanings to possession, the possess role is taught to associate different patterns with different meanings. For this sentence, the pattern is <category, category>. However, unlike the define role, the meaning for this pattern is not to create a hierarchical relation, but rather to add the aspect of occupation to the human category. A current list of patterns with their meanings for both the define role and the possess role are detailed in Table 1. This list is by no means complete and can be expanded as new patterns are introduced. Finally, we note that all these meanings were originally taught by a human teacher instead of pre-programmed.

TABLE I
LEARNED PATTERNS

Pattern	**Meaning**	**Example**
Define Role		
<unknown, kind>	Creates a new kind (category, idea, etc)	Countries are a category.
<unknown, category>	Creates an object of the category	Canada is a country.
<category, category>	Creates a hierarchy relationship	Apples are fruits.
<object, category>	Adds an attribute	John is upset.
<object, object>	Adds an attribute	Ted is a civil engineer.
<concept, concept value>	Adds a concept attribute	The height of Mt. Everest is 8,850 m.
Possess Role		
<category, concept>	Adds an attribute	Vehicles have weight.
<category, category>	Adds an aspect	Humans have emotions.
<object, category>	Adds a possession	John has 3 children.

Patterns and their meanings for the define and possess roles.

Once the final declaration thought has been understood completely, its intention can be carried out, when needed, by simply calling the existing functions with arguments set appropriately. However, there are some limitations to the executing capabilities of the define role. For example, when defining a new kind, only categories and ideas can be properly created; while other kinds of knowledge are not created currently. There are several reasons for that. The first reason is that the existing learning program requires a large amount of information to create knowledge objects of those kinds. The second reason is that the name of the sub-kind is needed in order to create the correct type of objects to preserve polymorphic behavior in an Object-Oriented solution. To further complicate the situation, each sub-kind may require different initial component knowledge. For example, consider learning the two concepts: mass and force. When teaching mass, a fundamental concept, the following information is needed: its type, name, an initial unit of measurement, any shorthand notation, and which system the unit belongs to such as the metric system. On the other hand, for learning force, a concept derived from other concepts, the formula of the product of mass and acceleration is needed. This amount of information is unlikely to be provided in one sentence and will generally require a discourse to gather the appropriate amount of knowledge before creating the correct knowledge object. Alternatively, the system could be set up to incrementally add to the definition of a knowledge object rather than requiring all the details up front.

VI. Discussion and Conclusion

We are currently working on using these learned language knowledge to produce a sentence reflecting a given internal thought, where examples on the creation of internal thoughts may be found in [37]. The sentence generation process is similar to the parsing processing. It uses the structures and rules of the grammar term to ensure a grammatically correct sentence, while the roles are for correlating the knowledge in the internal thought to the semantics of the various parts of the sentence. We are also working on identifying the knowledge associated with a pronoun before determining the structure of the noun phrase. One possible solution is to use the current context built up during a discourse. In addition, we are currently trying to solve the complications encountered when dealing with prepositional phrases. One complication is that, although a preposition is taught to associate with one structure, some prepositions may be used for more than one structure, e.g., ‘by’. With the phrase “by the bookshelf" a relationship structure should be created. On the other hand, the phrases "by Sunday" and "by himself" represent a time and a means; both should be represented by additional-information structures. A potential extension is to allow multiple structures for a preposition, but the problem now is to determine the correct structure when the preposition is used. One possibility is to use the context of the sentence to determine the structure since using the preposition alone is not sufficient. The next complication is that some prepositions may be used for multiple implications. In [37], prepositions such as 'for', 'to', and 'from' are taught to have the additional-information structure and are associated with logical objects in individual actions. For example for action buy, 'from' is associated with the seller, while 'for' is associated with the price of the item. As a result, effects of the action may be associated with the correct knowledge objects and recognized easily. So when given the sentence "John buys a gift for 10 dollars.", one of the effects that John spends 10 dollars can be easily recognized. Now, consider "John bought a gift for Mary for her birthday for 10 dollars." In this example, each prepositional phrase implies a different effect involving different knowledge objects. Once again, the preposition alone is not sufficient in determining the meaning or implication of the phrase. Therefore, the problem is to distinguish the different meanings of the prepositional phrases when the same preposition is used. Another common complication when dealing with prepositional phrases involves the ambiguity problem. With a sentence such as "I saw a man with a telescope", it is unclear whether the prepositional phrase should be attached to the noun, meaning the man was carrying a telescope, or to the verb, meaning I saw with a telescope. Such an ambiguity is difficult to solve via the grammar alone and one possible solution would require a reasoning agent to infer based on the context of the sentence.

An important requirement for an artificial intelligent system to communicate is its capability to understand the intention of a given sentence. We have shown in this paper that there are multiple levels of understanding that collectively give the overall intention of a sentence. The nouns, prepositions, noun phrases, and verb of a sentence all play integral roles. For nouns, their location in a noun phrase determines how they will be used in the overall knowledge structure of the noun phrase, while the preposition helps to determine the structure of that knowledge. For noun phrases, their position in the sentence defines their role in the overall intention of the statement. Finally, the verb determines the interactions and relationships between the noun phrases.

References

[1] R. Weischedel, J. Carbonell, B. Grosz, W. Lehnert, M. Marcus, R. Perrault, and R. Wilensky, "White paper on natural language processing," *Human Language Technology Conference*, Association for Computational Linguistics, Morristown, NJ, 1989, pp. 481-493.

[2] A. Gal, G. Lapalme, P. Saint-Dizier, and H. Somers, *Prolog for Natural Languages Processing*, John Wiley and Sons Ltd., 1991.

[3] C. Matthews, *An Introduction to Natural Language Processing Through PROLOG*, 1st edition, Longman Publishing Group, 1998.

[4] M. A. Covington, *Natural Language Processing for PROLOG Programmers*, 1st edition, Prentice Hall PTR, 1993.

[5] F. Pereira, "Logic for natural language analysis," SRI International, technical note 275, 1983.

[6] D.A. Turner, "A new implementation technique for applicative languages," *Soft. Pract. Exper.* vol 9, no. 1, pp. 31-49, 1979.

[7] P. Hudak, S.L. Peyton-Jones, P. Wadler, B. Boutel, J. Fairbairn, J.H. Fasel, M.M. Guzman, K. Hammond, J. Hughes, T. Johnsson, R.B. Kierburtz, R.S. Nikhil, W. Partain, and J. Peterson, "Report on the programming language Haskell, a non strict, purely functional language," *SIGPLAN*, vol. 27, no. 5, R1-R164, 1992.

[8] G. Huet, "A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger," *Journal Functional Programming*, vol. 15, no. 4, pp. 573-614, 2005.

[9] R. Leermakers, "The Functional Treatment of Parsing," *International Series in Engineering and Computer Science*, Kluwar Academic Publishers, 1993.

[10] P. Ljunglof, "Functional pearls: Functional chart parsing of context-free grammars functional pearl," *J. Funct. Program*, vol. 14, no. 6, pp. 669-680, 2004.

[11] P.C. Callaghan, *Generalized LR parsing,* The Happy User Guide, Chap. 3, Simon Marlow, 2005.

[12] R. Garigliano, R. Morgan, and M.Smith, "The LOLITA system as a contents scanning tool," in *Proc. 13th International Conference on Artificial Intelligence, Expert Systems and Natural Language Processing*, Avignon, France, 1993.

[13] R.A. Frost, "W/AGE the Windsor attribute grammar programming environment," in *Proc. IEEE Symposia on Human Centric Computing Languages and Environments*, 2002, pp. 96-99.

[14] R. A. Frost, "Realization of natural language interfaces using lazy functional programming," *ACM Computational Survey*, 38, 4, pp. 11, 2006.

[15] N. Chomsky, "Three models for the description of language," *IRI Transactions on Information Theory*, vol, 2, no. 3 pp. 113-124.

[16] P. Naur, J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J.H. Wegstein, A. van Wijnagaarden, and M Woodger, "Report on the algorithmic language ALGOL 60," *Communications of the ACM*, vol. 3, no. 5, pp 299-314,1960.

[17] N. Sager and R. Grishman, "The Restriction Language for Computer Grammars of Natural Language," *Communications of the ACM*, vol. 18, pp. 390-400, 1975.

[18] D. Knuth, "Semantics for context-free languages," *Mathematical Systems Theory*, vol. 2, pp. 127-145, 1968.

[19] C. Pollard and I. A. Sag, *Head-Driven Phrase Structure Grammar*, University of Chicago Press, Chicago, 1994.

[20] P. Kay and C. J. Fillmore, "Grammatical constructions and linguistic generalizations: The What's X Doing Y? construction," *Language*, vol. 75, no. 1, pp 1-33, 1999.

[21] A. K. Joshi, "Tree adjoining grammars: how much context-sensitivity is required to provide reasonable structural descriptions?" *Natural Language Parsing*, Cambridge University Press, Cambridge, pp. 206-250, 1985.

[22] I. A. Melcuk and A. Polguere, "A formal lexicon in the meaning-text theory (or how to do lexica with words)," *Computational Linguistics*, vol. 13, no. 3-4, pp. 261-275, 1988.

[23] J. Thorne, P. Bratley, and H. Dewar, "The syntactic analysis of English by machine," *Machine Intelligence*, vol. 3, pp. 281-31, 1968.

[24] Y. Bar-Hillel, "A quasi-arithmetical notation for syntactic description," *Language*, vol. 29, pp. 47–58, 1953.

[25] R. Montague, "Universal grammar," *Theoria*, vol. 36, pp. 373–398, 1970.

[26] M. Steedman, "Type-raising and directionality in combinatory grammar," in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berkeley, CA, 1991, pp. 71–79,

[27] L. Birnbaum and M. Selfridge, "Conceptual analysis of natural language," *Inside Computer Understanding*, 1981.

[28] A. Ranta, *Type-Theoretical Grammar*, Oxford University Press, Oxford, UK, 1994.

[29] C. Shan, "Monads for natural language semantics," in *Proc. 13th European Summer School in Logic, Language and Information*, Student Session, K. Striegnitz, Ed., Helsinki, Finland, 2001, pp. 285-298.

[30] A. Voutilainen, "A syntax-based part of speech analyzer," in *Proc. Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Dublin, 1995, pp. 157-164.

[31] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a Large Natural Language Corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313-330, 1993.

[32] W. Faris and K. Cheng, "An Object-Oriented Approach in Representing the English Grammar and Parsing," in *Proc. International Conference on Artificial Intelligence*, 2008, pp. 325-331.

[33] K. Cheng, "An Object-Oriented Approach to Machine Learning," in *Proc. WSES International Conference on Artificial Intelligence*, 2000, pp. 487-492.

[34] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, 2nd Ed., Prentice Hall, 2003.

[35] K. Cheng, "The Representation and Inferences of Hierarchies," in *Proc. IASTED International Conference on Advances in Computer Science and Technology*, 2006, pp. 269-273.

[36] K. Cheng, "Representing Definitions and Its Associated Knowledge in a Learning Program," in *Proc. International Conference on Artificial Intelligence*, 2007, pp. 71-77.

[37] E. Ahn, W. Faris and K. Cheng, "Recognizing the Effects Caused by an Action in a Declarative Statement," technical report #UH-CS-09-02, Computer Science Dept., University of Houston, February 2009.

[38] W. Faris and K. Cheng, "An Improved Syntactic Stage of a Parsing Algorithm," in preparation.