MEDWELL PUBLISHING
International Journal of Soft Computing 4 (2): 116-121, 2009
ISSN: 1816-9503

Semantic Representation of Natural Language Statements in Domain Description

[1]E.E. Williams, [3]J.O.A. Ayeni, [3]E. Fasina and [2]E.O. Ukem
[1]Department of Math/Statistics and Computer Sciences,
[2]Department of Physics, University of Calabar, Nigeria
[3]Department of Computer Science, University of Lagos, Nigeria

Abstract: When we learn a foreign language, much of our effort goes into the syntax or grammar. The same set of words with two slightly different ordering can mean completely different things and the rules for ordering are different in every language. For example, What does it taste like? and What taste does it like? are both meaningful but not the same. As Terry Winograd puts it, much of the modern linguistics has been devoted to analyzing the knowledge that underlies this ordering. Underlying all of the areas of knowledge about any language is a person's knowledge of the world, the objects in it and the relationships between them. It is necessary to understand how world knowledge and language understanding are interrelated. Naturally, we use the superficial linguistic processes involving words, terms/objects and structures to express world knowledge. In this study, we try to develop a method for representing and ordering objects and the semantic or meaning/relationship that could be given to such representation and ordering of the objects are provided. English language, as a natural language is used for illustration in the research.

Key words: Natural language, word ordering, semantics, semantic exploration, syntax, syntactic normalization

INTRODUCTION

Human beings are more comfortable to use natural language (a language whose rules are based on current usage without being specifically prescribed) for description of scenes/situations, categories of things in some domain and ultimately model the domain because the user does not need to learn any syntax or semantic of words of other languages and natural language is very flexible. The research load and completion times for such tasks are therefore, minimized. Bearing in mind these advantages, this study has used existing structural decomposition method to decompose domain into sub domains and provide a description of each sub domain, thus building a model of any chosen domain, using natural language to describe the concepts and objects in that domain.

A model is a simplified representation of certain aspects of the universe. Marker (2000) views the concept of models and model building as that which, concerns the construction of a formal theory that describes and explains the aspects. Therefore, we model a system or structure that we want to build by writing a description of it. A model is usually defined from a language text (L). The model gives an interpretation of the language text. For example, it is relatively easier to describe domain knowledge using natural language than to use a formal language. This research provides, semantic methodology for representing the natural language statements used to describe the domain in a machine understandable format. The approach is domain independent, but depends to some extent on the description logic used to describe the domain. Some aspects of ambiguity in natural language text and complex lexical structures are handled in the research by introducing syntactic normalization and semantic exploration. Thus, the development of an innovative, language independent semantic technology that supports the use of natural language for building domain model is achieved.

RULE ORDERING IN NATURAL LANGUAGE SENTENCES

In grammars with general rewrite rules, it is possible for the order in which, the rules are applied to have a significant effect on the outcome of derivation. Paul and Pulman (2006), is of the opinion that there may be a possible derivation in which, rule A is applied, followed by rule B. But if rule B is applied first, it may change the structure so that rule A is no longer applicable. As linguists examined the various transformations that were needed to describe English, they found tantalizing

Corresponding Author: E.E. Williams, Department of Math/Statistics and Computer Sciences, University of Calabar, Nigeria

regularities in the consequences of choosing particular orderings. There are sets of rules with a natural underlying order, which therefore must have some kind of psychological reality. For example,

$$\text{John loves Alice} \xrightarrow[\text{passive}]{} \text{Alice is loved by John}$$

and

$$\text{Okon ate fish} \xrightarrow[\text{passive}]{} \text{Fish was eaten by Okon}$$

have the same underlying structure. However, this type of example deals with only transitive verbs. This situation therefore calls the attention of parser designers to have a specific parsing order in mind. Normally, the parser is assumed to work from left to right and top-down. We shall use the issue of transitivity and subject verb agreement to demonstrate our method used in the analysis.

CLASSIFICATION OF WORLD LANGUAGES

The idea of rule/word ordering in grammars of languages discussed above has given rise to a way of classifying languages based on the word order used by the languages. The world languages can therefore be classified into Subject Object Verb (SOV), Subject-Verb-Object (SVO), Verb-Subject-Object (VSO) and Verb-Object-Subject (VOS), word arrangement (Natural Language Classification, 2006). Our method for semantic representation is applicable to the set of natural languages that have subject and predicate syntax (SVO structure).

DOMAIN SEMANTICS

Sheth and Ramakrishnan (2003) see semantics as the key ingredient in the next phase of the web infrastructure as well as the generation of information systems applications. The significance of semantics and the challenges they pose in developing semantic techniques are not new to researchers in database and information system field. Although, differences exist between domains, general agreements exist about several issues related with the structure and behaviour of real world objects as observed by Chandrasekaran *et al.* (1999):

- There are objects in the world
- Objects have properties or attributes that can take values, i.e., they can be represented as triplets (Object → Attribute → Value)
- Objects can exist in various relations with each other
- There are processes that occur over time, in which objects participate. The world and its objects can be in different states

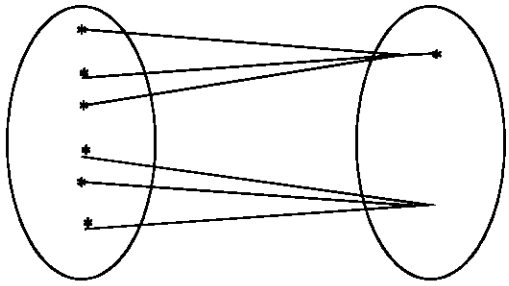

Fig. 1: Many to one mapping of objects-functions

In this research, we use the terminology object or term inter-changeably. We can define a term as:

- Individual constant and individual variables. By constant, we mean names of individual objects while, variables are names of groups of objects
- If we have terms $t_1, t_2 \ldots t_n$ and there is a function f of n variables, then $f(t_1, t_2 \ldots t_n)$ is also a term. That is, a function is a mapping of objects/variables to other objects. It could be many to one relationship

For example, we use the following to refer to terms or entities in a system:

- Directly by name (constant) e.g., Joe
- A group of terms by using group name e.g., girls, variable, men
- A function or property or value of an individual/group e.g., ID-No. or Reg. No. A function is an assignment of a value to situations like from range to each domain point. Reg. No. in this case is a function of group of students and has a 1:1 mapping (Fig. 1)

We shall use these ideas for semantic representation in our method.

DESCRIPTION OF THE PROPOSED SEMANTIC REPRESENTATION OF NATURAL LANGUAGE STATEMENTS IN A DOMAIN DESCRIPTION

Based on the analyses above, our study at this stage involves writing down a list of all terms, we would like either to make statements about in a domain or to explain to a user. What are the terms we would like to talk about? Suppose we use the activities taking place in a river estuary as domain to build a model. Initially, it is important to get a comprehensive list of terms/objects in the domain without worrying about overlap between concepts, which they represent, relations among the terms, or any properties that the concepts may have.

Table 1: An example of natural language (English) description of a scene

Sample sentences	Terms	Predicates
The fisher eats soup	Fisher, soup	0 *eats*
Akpan eats yam	Akpan, yam	
The customer eats fish	Customer, fish	
The fish eats meat	Fish, meat	
The fisher lives in a hut	Fisher, hut	1 *lives in*
Okon lives in a fishing-village	Okon. fishing-village	
Fish lives in water	Fish, water	
Akpan lives in the town	Akpan, town	
The fisher drinks palm-wine	Fisher, palm-wine	2 *drinks*
The customer drinks gin	Customer, gin	
Akpan drinks whisky	Akpan, whisky	
A large-boat is driven by outboard engine	Large-boat, outboard engine	3 *is driven by*
Motor-car is driven by auto-engine	Motor-car, auto-engine	
A trawler is driven by diesel-engine	Trawler, diesel-engine	
14 sentences	22 terms	

Table 2: Terms and corresponding attributes

Attr. 0:	Attr. 1:	Attr. 2:	Attr. 3:	Attr. 4:	Attr. 5:	Attr. 6:	Attr. 7:
Fisher	Soup	Fisher	Hut	Fisher	Palm-wine	Large-boat	Outboard-engine
Akpan	Yam	Okon	Fishing-village	Customer	Gin	Motor-car	Auto-engine
Customer	Fish	Fish	Water	Akpan	Whisky	Trawler	Diesel-engine
Fish	Meat	Akpan	Town				

We use an example to explain the basic methodology, we shall use. The example is based on a set of sentences used to describe the domain. Table 1 gives the tabulation of the sentences, the collection of terms in the sentences and the associated predicates (A predicate is the action word and links 2 terms).

We had earlier observed in the word rule ordering section that the position of the term in a sentence determines its role in the sentence. Therefore, it represents an attribute or symbol and is used to describe an instance of the term. Let us consider each of the attributes in the example. The term that occupies the first position acts on the term that is in the second position if we use active speech. A passive speech takes the reverse order. We selected 4 predicates (enclosed in *) using the set of sentences in Table 1. Each predicate corresponds to an entry to a lexicon in the domain. A binary relation maps instances of a class to another class.

Since, all our predicates are capable of linking 2 terms, we add eight attributes in this case (i.e., 4 predicates × 2 because each term can occupy either first or second position) to each term. We then form a table that holds the terms and their corresponding bit fields or attributes. These attributes are represented by binary digits (bits). We refer to each attribute position as bit field. Each position where the term appears in a predicate is represented with 1 in the table otherwise 0.

Let, us consider the tabulation in Table 1 and represent a term (where, it is linked with another term by a predicate) with 1 and 0 otherwise as stated above. For the term fisher the entries are as follows: We remember that we are dealing with binary predicates, hence, 2 bit positions are used for each predicate. For predicate 0, *eats*, we put 1 in the first bit position for fisher. The second predicate, 1 *lives in *, also has 1 for fisher in the first bit position. Similarly, the third predicate, 2 * drinks*, also 1 is entered in the first bit position for fisher. Attributes 6 and 7 each contain 0 because the fourth predicate in our illustration has nothing to do with fisher. Considering each of the attributes and the corresponding bit entries, the action gives Table 2 and 3.

We wish to state here that bit sequence is universally known. Therefore, if an operation or relationship is converted to bit sequence, such operation or relationship can be interpreted universally. What do we do with the Tables and in particular Table 3.

Some very useful results can be obtained by examining the bit patterns in Table 3 in particular as follows:

We had earlier stated that a binary relation maps instances of a class to another class. If we collect out those terms, which have 1 in an attribute, for all attributes, we are forming groups. We note here that the same bit sequence implies same group of objects/concepts. This is true irrespective of the language used for the description of the domain because objects/concepts do not change with language. This therefore, makes this study language independent.

Looking at the groups obtained, it can be seen that these groups form concepts as follows:

From our Table 3, attribute 1 represents food. Attribute 0 represents those that eat the food. Attributes 2 and 3 represent occupants and accommodations. Attributes 4 and 5 represent drinkers and the drinks. Finally, the groups based on attributes 6 and 7 represent objects that can be moved and objects used for that movement.

Table 3: Terms and their corresponding bit fields

	Bit positions↓									Bit Positions↓								
Terms↓	0	1	2	3	4	5	6	7	Terms↓	0	1	2	3	4	5	6	7
Fisher	1	0	1	0	1	0	0	0	Soup	0	1	0	0	0	0	0	0
Akpan	1	0	1	0	1	0	0	0	Yam	0	1	0	0	0	0	0	0
Customer	1	0	0	0	1	0	0	0	Fish	1	0	1	0	0	0	0	0
Fish	1	0	1	0	0	0	0	0	Meat	0	1	0	0	0	0	0	0
Okon	0	0	1	0	1	0	0	0	Hut	0	0	0	1	0	0	0	0
Water	0	0	0	1	0	0	0	0	Fishing village	0	0	0	1	0	0	0	0
Town	0	0	0	1	0	0	0	0	Palm-wine	0	0	0	0	0	1	0	0
Large boat	0	0	0	0	0	0	1	0	Gin	0	0	0	0	0	1	0	0
Out board engine	0	0	0	0	0	0	0	1	Whisky	0	0	0	0	0	1	0	0
Motor-car	0	0	0	0	0	0	1	0	Auto-engine	0	0	0	0	0	0	0	1
Trawler	0	0	0	0	0	0	1	0	Diesel engine	0	0	0	0	0	0	0	1

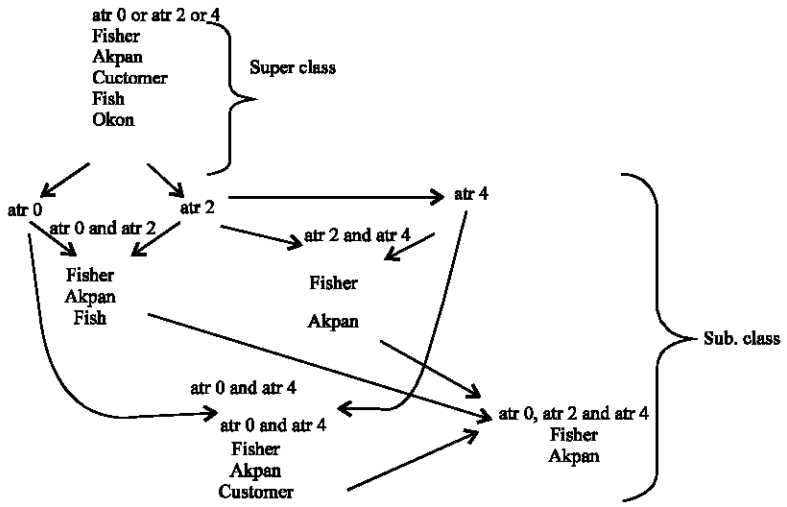

Fig. 2: Illustration of class and super class hierarchy

By examining, the bit patterns in the table (of course implementation of these concept will involve a table larger than this) the embedding of objects and language homomorphism can be established. The overlapping among groups in tables of this nature clearly gives the embedding of groups into another group. Also, studying the tables we can establish axioms, which are true of the model. We always have in mind that a binary relation maps instances of a class to instances of another class as proposed by Meersman (1999). This type of grouping therefore defines class, sub-class, or super class hierarchies and the listed instances are the instantiations of the model. If an object has independent existence rather than describe other objects, such an object forms a class in the domain and will become anchors in the class hierarchy. Noy and McGuiness (2001) is of the opinion that we can view classes as unary predicates (they answer or are questions that have only one argument). If A is a super class of class B, then every instance of B is also an instance of A. In other words, the class B represents a concept that is a kind of A. In our example, we combine 2 or more attributes using the 'and' operator to form a sub-class. We also, combine or form a super class by using the OR (V) operator to combine sub-classes. Using the example and following the above rules, we can see that attr 0 and attr 2; attr 2 and attr 4; attr 0 and attr 4 will form sub-classes. attr 0 or attr 2 or attr 4 will form super class. For example, we can say that it may not be only Akpan that eats Yam but we can have a set of Yam eaters. The situation, showing the concepts that exist in the domain, can be expressed as shown in Fig. 2.

IMPLEMENTATION CONSIDERATIONS FOR TRIARY PREDICATES AND HIGHER

The illustration shown above shows what happens when binary predicate is used. For triary predicates and above, similar conditions will appear, except that the table increases in size to accommodate the terms and associated attributes. For example:

Example 2: One may also have a list of triples-Triary of terms in a sentence, for example, a sentence such as Akpan eats fish at home. Here, the sentence is made up of 3 terms-Akpan, fish and home.

Example 3: One may have yet a list of quadruples-Tertiary. An example of this is a sentence such as Akpan eats fish at home in the veranda.

Here, the sentence is made up of 4 terms-Akpan, fish, home and veranda. Consequently, we may have a list of k-ary terms in a sentence.

A predicate, as we saw earlier, expresses relationship among terms. In examples 2 and 3, eats represents predicate or shows relation between Akpan and fish. Similarly, in the expresses relation between veranda and home. As indicated earlier, the sizes of tables increase with this number of terms and compound sentences may contain variable number of terms. Working with compound sentences that give rise to triary terms and higher will therefore, increase the sizes of tables generated. To avoid creating unnecessarily large tables, we make the following assumptions and study towards solving the problem:

Assumption: Given a sufficiently large, well-conditioned (simple sentences) and a random system of sentences, it is possible to identify all the terms in the given text. With this assumption in mind, we approach the problem as described in syntactic normalization and semantic exploration in natural language text section.

SYNTACTIC NORMALIZATION AND SEMANTIC EXPLORATION IN NATURAL LANGUAGE TEXT

People often express their thoughts with complex sentences. This complexity makes human communication more efficient by reducing redundant phrases. Such eloquence is certainly appropriate for ordinary rhetoric. However, to make sentences suitable for modeling, we need to greatly simplify their syntax, often at the expense of some apparent redundancy (e.g., repeatedly naming a subject in several related sentences). The above assumption therefore, means that we must:

- Develop method of writing precisely for the study
- Possibly use a restricted natural language, which is inherently unambiguous and more precise

We use transformations that can be applied to sentences to preserve their overall meaning while, producing simple and consistent syntactic formats. Collectively, we can characterize these transformations as syntactic normalization. Simplifying the syntax of sentences makes them less prone to ambiguity. Such simplicity can also help people to share and compare their mental models.

Sometimes a complex sentence cannot be simplified to the desired degree without first exposing the meaning of some of its constituent phrases. In this type of situation, semantic exploration may be necessary to provide clues for recovering nouns or verbs from descriptive adjectives and adverbs and other kinds of phrases.

Normalization, in database parlance, is the process of removing undesirable features by breaking a relation into other relations of desirable structures. It is a step-by-step reversible process of transforming unnormalised relation into relation of progressively simpler structure. Since the process is reversible, no information is lost during the transformation. The system ignores extraneous, irrelevant information. Normalized sentences are therefore simple declarative sentences, sometimes called kernel sentences.

Actually, natural language processing in this area of application is yet to reach the point where open-ended usage is possible. Naturally, any natural language regenerates and creates new words based on world activities/scenes. The realities of computer software and hardware limitations such as memory, in turn impose restrictions on natural language capabilities. Chinmezie and Ogbuji (2000) is of the opinion that the realistic amount of data necessary to perform natural language processing at the human level requires a memory space and processing capacity that is beyond even the most powerful computer processors.

REPRESENTATIVE EXAMPLE OF SEMANTIC NORMALIZATION AND EXPLORATION

The following brief narration can be used to illustrate, the concept of semantic normalization and exploration.

> The fish are stored in the hut; in the compound there are also huts that house old and new fishing nets and fishermen. Each hut is allowed to hold a maximum number of fish.

The sentences describe the hut buildings. Several applications of syntactic normalization are needed to simplify the sentence and split them into its constituent clauses. A mix of syntactic normalization and semantic exploration is needed to reveal the ideas contained in the sentence. We can analyze the sentences as indicated in Table 4. The asterisks (*) in the table indicate those sentences that should be retained in the final domain model. The set of sentences in domain model are those that we use in creating tables as discussed in our proposed semantic representation of natural language statements.

Table 4: Example analyses

Convert verb to active voice	The hut stores the fish (This convert ion is optional since our system can handle passive voice as discussed in section 5.2)
Convert subject to singular	*A hut stores some fish.
Verb isolation	Huts house old and new fishing nets
Simple generalization	Old and new fishing nets = fishing gear
Convert subject to singular	*A hut houses fishing gear.
Verb extraction-'in the compound'	*Compound contains some storage huts *Compound contains dwelling huts *A fisherman lives in a hut

CONCLUSION

The method developed in this research provides a means of representing sentences and the associated semantic interpretation in domain descriptions. Sentences that have subject-verb-object are used for illustration. However, some modifications can permit the method to be applied to other language structures such as SOV, VSO and VOS. The considerations in these cases will involve the re-positioning of terms and hence, the bit arrays. By doing this, the method can be applied universally because it considers only terms and relations provided by predicates. Terms in domain descriptions are unique and do not depend on languages used to describe them.

The method has several potential areas of application such as text analysis and in building systems such as ontology that may need heterogeneous agents to access them.

REFERENCES

Chandrasekaran, B., J. Josephon and V. Benjamins, 1999. What are ontologies and why do we need them? IEEE. Intelligent Syst., 14 (1): 20-26.

Chinmezie and T. Ogbuji, 2000. The future of natural language processing. Unix Insider, 8 (2): 23-27.

Marker, D., 2000. Introduction to model theory. University of Illinois, Chicago, MSRI Publications, 30 (2): 26-37.

Meersman, R., 1999. Semantic Ontology Tools in the Information System Design. Proc. ISMIS 99 Conf. Spring Verlag Publishers, pp: 126-130.

Natural Language Classification, 2006. Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php.

Noy, N.F. and L.D. McGuiness, 2001. Ontology development 101: A guide to creating your 1st ontology. Standard Knowledge Systems Laboratory Technical Report KSL-01-05 and standard Medical Informatics Technical Report SMI-2001-0880, Noy@smi.standard.edu, dlm@ks1.standard.edu.

Paul, D.J. and S. Pulman, 2006. Sentence ordering with manifold-based classification in multi-document summarization. Proc. 2006 Conf. Empirical Methods in Natural Language Processing (EMNP), Sydney, pp: 526-533.

Sheth, A. and C. Ramakrishnan, 2003. (Wed) Technology in Action: Ontology Driven Information Systems for Search, Integration and Analysis. IEEE Data Eng. Bull., 2 (3): 56-62.