A unified approach to natural language treatment

Jordi Àlvarez

Report LSI–98–10–R

A Unified Approach to Natural Language Treatment

Jordi Alvarez
Universitat Politècnica de Catalunya
Departament de Llenguatges i Sistemes Informàtics
Mòdul C6, Campus Nord, 08034 Barcelona (Spain)
jalvarez@lsi.upc.es

Abstract

This paper proposes a learning-based methodology to deal with Natural Language. Our system tries to acquire a knowledge model from an analyzed text. After that, a new text can be analyzed by using that model.

The methodology introduces a unified representation mechanism for every kind of information present in Natural Language treatment (corresponding to Natural Language phases in other systems). In this way, the acquired knowledge model can contain any kind of language information (lexical, syntactic, semantic, conceptual or even world-knowledge).

With this approach we are not trying to give an overall solution to Natural Language treatment; but to solve Natural Language problems using any kind of relevant information.

We also provide an inference mechanism that puts asside the idea of Natural Language phase.

1 Introduction

Traditionally, Natural Language treatment distinguishes among several phases (lexical, syntactical, semantical/conceptual, contextual ...). This phases could be though as subproblems. Most Natural Language system try to give a solution to one of these subproblems by using only information from that subproblem. Systems that try to give a solution to all or several phases mostly attack the problem by boarding each phase independently and passing the results to the next phase.

From a cognitive point of view, humans don't seem to have these *tools* that perform independent calculations and, working in *pipe-mode*, pass information one to the next.

This does not mean this kind of approach is an incorrect one. What we want is to solve the problem; and not necessarily to solve the problem exactly as humans do. But what is clear to me is that there exist correlations among information from what we have decided to distinguish as different phases.

As we can find much more correlations among information from the same phase, I understand that the *phases approach* is a simplified model for Natural Language treatment. This simplification could work when interactions among different phases are not important. Most of the systems existent to date use this approach and give acceptable results.

But if we want to go further and improve these results, we have to take into account interactions among information from different phases [10]. Probably, this will also reduce the size training set in order to obtain good results; and it also would reduce the complexity of the acquired model (since a lot of special cases may refer to interrelations with other kinds of information; and so, they would disappear as special cases). By other hand, we should not forget that taking into account these interactions can make our problem unapproachable due to the amount of resources needed to solve it.

Although there exist some systems that use different kinds of information to solve "better" some Natural Language problems, the interaction they allow to express is quite restricted. This paper proposes an overall Natural Language model in which we will be able to reason about interactions among any kind of information.

The representation formalism we have chosen to represent the overall Natural Language model is YAYA [3]; a restricted Description Logics system focused on automatic acquisition of knowledge models from examples. YAYA has a learning algorithm that performs an heuristic local search over the space of possible concept definitions. The knowledge model acquired by the learning algorithm represents a set of correlations among Natural Language information. Once a model has been acquired, it can be used to infer knowledge over a new set of examples (that is, analyze a new text). This is further explained in next section; and to know more about YAYA, you can take a look at [3].

2 YAYA: The Underlying Representation and Reasoning Formalism

YAYA has been designed to be a general learning-reasoning tool for complex domains; that is, domains in which there appear complex relations among objects.

The *semantic network* formalism is the basic YAYA representational element. Restricted Description Logics capabilities have been added to the semantic network formalism. These capabilities allow to:

- Define concepts in terms of other concepts and relations among them.
- Decide when a concept is subsumed by another concept (work done by what has been called terminological reasoner or TBox in [6]); and when an object is an instance of a given concept (work done by what has been called assertional reasoner or ABox in [6]).
- State a set of properties the instances of a concept have. These set of properties are called *incidental properties* by [5]; and are different from the concept definition, that is a necessary property.

For every concept that has been defined, the set of incidental properties referent to it conforms the *concept description.* For every incidental property, YAYA keep the ratio of instances of the concept that have that property. From now on we will call these properties description properties.

2.1 Concept Definition Language

The concept definition language (CDL) is the language we use to state properties that objects have. These properties can act as concept definitions (stating that every object that has the property will be an instance of the concept) or description properties (stating the probability that an instance of a given concept has a property).

YAYA CDL is quite a restricted language compared to other Description Logics systems such as LOOM [7] or Classic [5]. It has been designed this way because of two reasons:

- We wanted to reason only with what we have in a semantic network: nodes and relations.

  Moreover, we take the position that knowledge is always and only at the concept level. That is, we can only distinguish the behaviour of two objects if they are instances of different concepts.

- YAYA CDL will be used by the learning algorithm to define concepts and to define new description properties. The more complex is the CDL used by the algorithm the bigger is the cardinality of the set of possible terms expressable by that language. This way, the learning algorith would spend much more time in deciding which terms to define and which ones not to define. So, we want to keep the language as simple as possible (the language must be expressive enough to allow representing a good knowledge model of the data).

Because of this, properties stated in the YAYA CDL only can state being an instance of a concept and being related to other objects that are instances of other concepts.

For example, we can state: (: *strc man* $\overset{married\text{-}to}{\longrightarrow}$ *person*). This property could be the definition of *husband.* We can go further and we can state: (: *strc man* $\overset{married\text{-}to}{\longrightarrow}$ *person* $\overset{was\text{-}married\text{-}to}{\longrightarrow}$ *person*); representing the set of husbands that are married to a person that had been married before marrying them.

With the *:strc* construct we can state chains as large as we want. YAYA CDL also has the constructs *:and* and *:or*, that allow stating conjunctions and disjunction among properties.

Finally, the language also has the inner construct *:unify*, that allows to bind nodes to variables. This allows to state simple things as the simmetry of the *married-to* relation:

(: *strc* (: *unify* person: x) $\overset{married\text{-}to}{\longrightarrow}$ *person* $\overset{married\text{-}to}{\longrightarrow}$ (: *unify* person: x))

Or more complex things as in the next property, that states for the instances having it that the parent of the parent is also the grandparent:

$$(:\ and\ \ (:\ strc\ \ person \xrightarrow{has\text{-}parent} person \xrightarrow{has\text{-}parent} (:\ unify\ \ \text{person: } x))$$
$$(:\ strc\ \ person \xrightarrow{has\text{-}grandparent} (:\ unify\ \ \text{person: } x)))$$

2.2 Learning

YAYA CDL, although quite restrictive, allows to state a lot of different properties. Properties can be concept definitions or description properties (these two sets are not necessarily disjoint). As we cannot represent every possible property, we must decide which concepts and description properties will be represented. This work is done by the learning algorithm.

The learning algorithm performs a local search around the concepts already defined in the knowledge base (search as learning was introduced by [9]; and in fact, a lot of current machine learning algorithms perform local searches over the space of possible theories [11]). We can start from scratch (with a root concept), with a set of predefined concepts or metaconcepts or with an incomplete knowledge model acquired manually or whatever.

The learning algorithm is an iterative process that reacts to every event that happens in the knowledge base (for example, the creation of a new concept). It consists of a set of operators that apply to these events.

The process is guided by a set of examples that are supposed to contain no erroneous properties. The algorithm has designed to be quite robust; some tests with *noisy synthesized data* have been performed and the obtained result has been succesful.

Learning is also guided by a set of heuristics. An heuristic value is given to the application of an operator. The potential knowledge generated by an operator is also given an heuristic value. Among the heuristics used, we can find:

- Some heuristics that restrict the application of operators or the creation of a new concept.
- Another is based on the idea of *concept utility*, that gives a representative value of the set of useful inferences that the system will be able to do with the definition of the concept.
- Others are computed from the similarity between concepts[1] and the amount of knowledge that a concept still does not predict (so as to go on with the search in order to find specializations of the concept that predict that knowledge).

The learning process is explained in more detail in [3].

2.3 Inference

Once a knowledge model has been acquired from a set of correct examples (for Natural Language this set of examples would correspond to an analyzed text); we can use that

[1]Similarity between concepts is computed as the amount of common information that both concepts have. A more detailed explanation can be found in [3].

model to infer knowledge over a new set of incomplete data (for Natural Language this new set would be a text that has not been analyzed).
The inference process is an iterative process in which:

1. Objects representing the new set of data are classified automatically by the YAYA ABox into the acquired model hierarchy.[2]

2. For every one of those objects, we look at the description of their classes[3] and get the properties with a probability over a confidence ratio.

   If the object hasn't those properties, they are put on the inference queue with a priority equal to the property probability for that object.

3. If the inference queue is empty, the inference process is finished.

   Otherwise, take the first element from the inference queue, infer it in the knowledge base and go to the first step.

In order to enhance the inference process, YAYA introduces probabilistic links between nodes. So, we can infer properties with lower probabilities introducing probabilistic links. The inference process will evolve the probability value of those links; and at the end of the inference process YAYA will determine those links that are true and those that are false.

Notice that, for Natural Language, this inference process banishes the idea of Natural Language phase. The process performs inferences ordered by a probability that YAYA assigns to it; and this order is independent of the kind of information involved in them.

3 Natural Language Model

This section is an outline of a general model for Natural Language. The model explained here includes information of different kinds. The purpose of the model is to be able to represent a whole Natural Language text analysis; and specially to make easy the capture of interelations among different kinds of information. Depending on the nature of the Natural Language problem we want to solve, the model can be adapted, discarding or modifying some kinds of information.

We want to represent the text and also the analysis of the text into a semantic network. In order to do it, the Natural Language model will provide a set of concepts and metaconcepts (see figure 1) and a set of relations (see figure 2). This will constitute the basic model that YAYA learning process will complete.

Our intention with this basic model is only to define the different kinds of information that will appear in the Natural Language analysis and the relations between them. Because

[2]This classification is performed automatically by YAYA. It is explicitly stated in the algorithm in order to notice that classes of objects (see next footnote) may change when an inference is performed.

[3]Object classes are the set of the most specific concepts that have one object as instance. As YAYA allows multiple inheritance and, in fact, the learning process builds hierarchies with multiple inheritance, an object may be associated to several classes.

of this, we can think of considering it a *metamodel.* Some of the concepts provided by the model will act at a concept level and some of them will act at a metaconcept level.

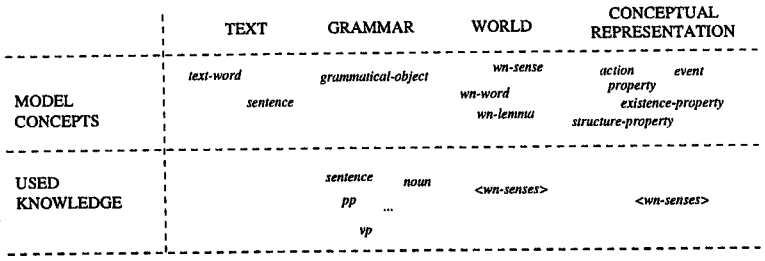

Figure 1: The concepts of the Natural Language model and the knowledge that can be used in the learning process and to perform inferences.

The model provides the following kinds of information:

Textual information The first element that has to be represented is text. Basic objects defined by the model are: *text*, *sentence*, *text-word* and *punctuation-sign.*

Instances of these concepts are interrelated between them with the relations: *next*, *previous* and *has.*

Word senses and world knowledge Regarding word sense information, YAYA takes the information from *WordNet* [8], a widely used linguistic ontology. YAYA accesses *WordNet* internally. In this way, we can view *WordNet* elements as a YAYA semantic Network[4].

To represent *WordNet* information as a part of our Natural Language model, we provide three concepts:

wn-word corresponds to the words in the dictionary.

wn-lemma refers to the lemmas corresponding to the words.

wn-sense has as instances all the senses corresponding to the lemmas.

Senses are related among them with the relations: *hyperonim*, *hyponim* and *synonim.* In the future we plan to introduce new relations among senses (*part-of* ...)[5].

Lemmas are defined to be instances of senses[6]. This is important to the learning algorithm, that will be able to use senses information (as they are concepts).

Finally, *text-words* are related to the corresponding *wn-word* with the *wn* relation.

[4]In the future, the use of other ontologies such as the Generalized Upper Model (GUM) [4] will be studied in order to substitute or complete the information provided by *WordNet*; either here or at the conceptual representation level (see below)

[5]Although WordNet has these relations, YAYA's representation of WordNet hasn't them yet. It can be interesting to make a study of the kind of inferences they can help to perform.

[6]This makes *wn-sense* to be a metaconcept.

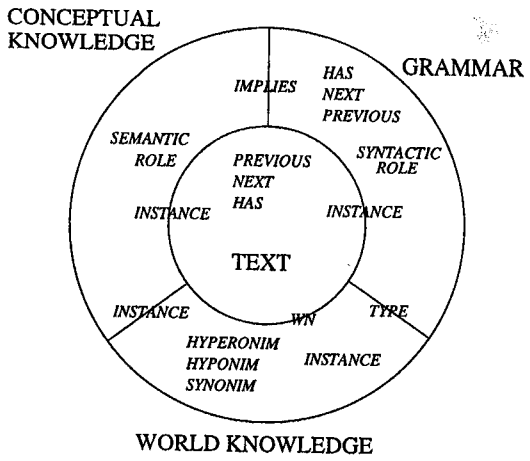

Figure 2: Set of relations of the model structured by kind of information they relate.

Grammatical information The representation of grammatical elements that will be used is quite simple. The model defines a concept for every grammatical category (*noun*, *verb*, *preposition*, *np*, *vp*, *pp*, etc). Each one of these concepts will be instance of *grammatical-object*.

Grammatical elements will be related among them with the *next*, *previous* and *has* relations. In addition, the model has a set of relations corresponding to roles that play some grammatical elements inside other grammatical elements (these relations can be seen as a specialization of *has*). In figure 3 we can see a syntactical analisis of a simple sentence.

The *text-words* correspond to simple grammatical categories. So, they are defined to be instance of them. By other hand, *wn-senses* also correspond to simple grammatical categories; so they are also defined to be instance of them. Notice that the *text-word* corresponding to a *word-sense* will always be instance of the same grammatical element that the *word-sense* is.

Semantic/conceptual representation To represent text meaning, we take the general model that appears in [1]. This model is taken as a set of guidelines; so that the conceptual representation model can be adapted to the problem to be solved.

Our model has five basic concepts: *conceptual-object*, *action*, *event*, *property* and *state*. Instances of these concepts are related among them with a set of *semantic-roles* (*agent*, *patient*, *instrument*, etc).

Regarding the interrelations with other kinds of information, the instances of grammatical elements (that is, every *np*, *noun* ... appearing in the text) are related to the corresponding *conceptual-objects* with the *has-representation* relation.

At this point, a world knowledge taxonomy could be very useful to give meaning to the *conceptual-objects*. For that purpose, we use again the *WordNet* senses taxonomy. Every *conceptual-object* will be instance of the corresponding *WordNet* sense.

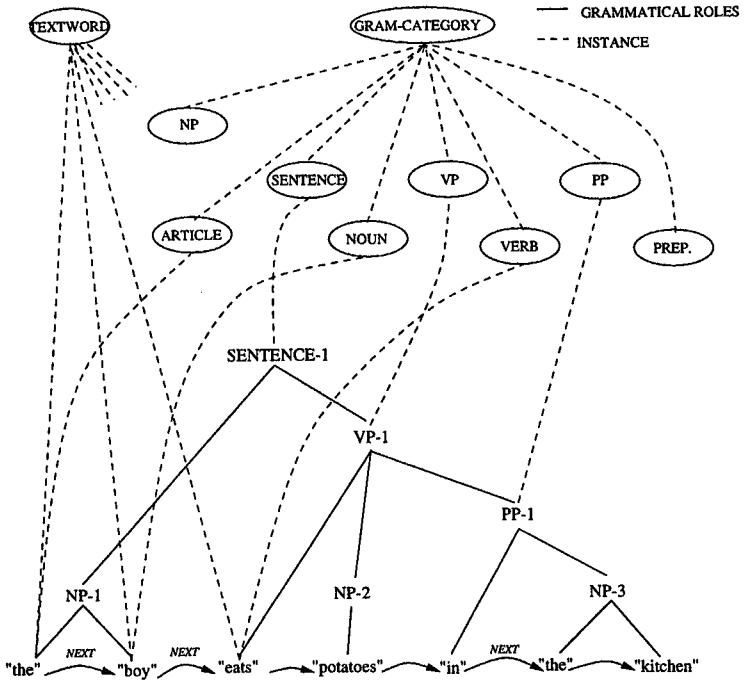

Figure 3: Example of syntactical analysis of a simple sentence.

Contextual information At this moment, the model does not deal with contextual information. But, depending on the problem to be solved, it could be quite useful to include contextual information in the overall Natural Language model.

In figure 4 we can see the piece of the semantic network corresponding to an appearence of the word table in a text. As we can see in the picture, the model makes an intensive use of the *instance* relation.

This extensive use of *instance* could be considered an abbuse or a heresy. It can make the same concept to appear as a *first-level* concept and as a *second-level* concept or metaconcept.

The answer I would give to this apparent problem is that the different kinds of information we are trying to model share some knowledge. And the best (and simplest) way to express this knowledge is through the *instance* relation; although it could seem heretic.

By other hand, this intensive use of the *instance* relation makes richer the knowledge available to the learning process without making it more complex.

4 Conclussion

YAYA has been implemented using Java. Its learning algorithm has been tested on some *artificially created examples* with some hundreds of instances. Test results have been succesful. In all the tests, instances have been generated artificially from a previous model. Once generated, this model has to be learned by YAYA.

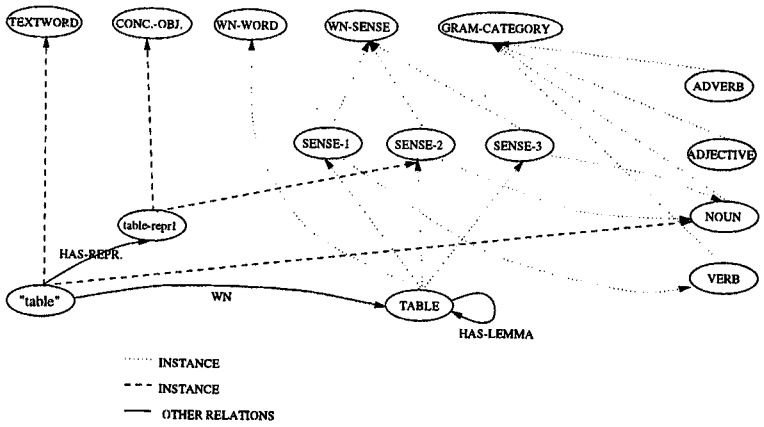

Figure 4: Representation of an occurrence of the word "table" in the text; and the relation that it has with sense, grammatical and conceptual knowledge (supposing this occurrence of table corresponds to the second WordNet sense for the word table).

The first experiments have been promising. Although the hierarchies and knowledge to be learned were quite simple, they have been designed to reproduce every possible situation the learning algorithm will found in trying to acquire a knowledge model from a Natural Language text. A report describing these experiments is in preparation and will be accessible very soon [2].

We are now starting to do Natural Language experiments. The first one will be that of [13], about semantic grammars. It has been formalized with our Natural Language model as containing lexical and conceptual knowledge (to which YAYA will add the knowledge provided by WordNet). After this, we expect to test the system on different Natural Language problems; for instance: word sense disambiguation and acquisition of selectional restrictions among others.

In conclussion, this paper proposes a machine learning-based methodology to deal with Natural Language. We first acquire a model from an analyzed text and then we use Description Logics classification facilities to implement an inference system that is able to analyze other texts.

The main point of this methodology is that it does not distinguish among different Natural Language phases. This allows to order the inferences to be performed by a probability ratio; making more robust the inference/analysis process.

We provide a Natural Language model. This model is represented as a YAYA semantic network. The model allows to represent lexical, syntactical, conceptual and world knowledge (and it could be extended for contextual info ...). Specially, it defines relations among these kinds of information trying to make easy the work of the learning algorithm that has to acquire the model.

References

[1] J. Allen. *Natural Language Undestanding.* The Benjamin Cummings Publishing Company, Inc., second edition, 1995.

[2] J. Alvarez. Basic experiments to test and design a local search learning algorithm. Technical Report in preparation, Universitat Politècnica de Catalunya, 1998.

[3] J. Alvarez. YAYA: Learning to reason in complex domains. Technical report, Universitat Politècnica de Catalunya, 1998.

[4] J. Bateman, B. Magnini, and G. Fabris. The generalized upper model knowledge base: Organization and use. In N. Mars, editor, *Towards Very Large Knowledge Bases*, pages 60–72. IOS Press, 1995.

[5] R. Brachman, D. McGuiness, F. Patel-Schneider, L. Resnik, and A. Borgida. Living with Classic: When and how to use a KL-ONE-like language. In Sowa [12], pages 401–456.

[6] R. MacGregor. The evolving technology of classification-based knowledge representation systems. In Sowa [12], pages 385–400.

[7] R. MacGregor. Using a description classifier to enhance deductive inference. In *Proceedings of the 7th IEEE Conference on Artificial Intelligence.* IEEE CS Press, 1991.

[8] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on WordNet. *International Journal of Lexicography*, 1991.

[9] T.M. Mitchell. Generalization as search. In J. Shavlik and T. Dietterich, editors, *Readings in Machine Learning*, pages 96–107. Morgan Kaufman Publishers, 1990.

[10] Ll. Padro. *A Hybrid Environment for Syntax-Semantic Tagging.* PhD thesis, Universitat Politècnica de Catalunya (UPC), 1998.

[11] J.R. Quinlan and R. M. Cameron-Jones. Oversearching and layered search in empirical learning. In *Proceedings of 14th International Joint Conference on Artificial Intelligence*, pages 1019–1024. Morgan Kaufman Publishers, 1995.

[12] John F. Sowa, editor. *Principles of Semantic Networks. Explorations in the Representation of Knowledge.* Morgan Kaufman Publishers, 1991.

[13] J. Zelle and R. Mooney. Learning semantic grammars with constructive inductive logic programming. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 817–822. AAI Press/MIT Press, 1993.

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports – 1998

LSI–98–1–R "Optimal Sampling Strategies in Quicksort and Quickselect", Conrado Martinez, Salvador Roura.

LSI–98–2–R "Query, PACS and simple-PAC Learning", J. Castro and D. Guijarro.

LSI–98–3–R "Interval Analysis Applied to Constraint Feasibility in Geometric Constraint Solving", R. Joan-Arinyo and N. Mata.

LSI–98–4–R "BayesProfile: application of Bayesian Networks to website user tracking", Ramón Sangüesa and Ulises Cortés.

LSI–98–5–R "Some reflections on applying Workflow Technology to Software Process", Camilo Ocampo and Pere Botella.

LSI–98–7–R "Trust Values for Agent Selection in Multiagent Systems", Karmelo Urzelai.

LSI–98–8–R "The use of SAREL to control the correspondence between Specification Documents", Núria Castell and Àngels Hernández.

LSI–98–9–R "Intervalizing colored graphs is NP-complete for caterpillars with hair length 2", C. Àlvarez, J. Diaz and M. Serna.

LSI–98–10–R "A unified approach to natural language treatment", Jordi Alvarez.

Hardcopies of reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6
Jordi Girona Salgado, 1–3
08034 Barcelona, Spain
secrelsi@lsi.upc.es

See also the Department WWW pages, http://www-lsi.upc.es/