Abstract

The technology needed for building intelligent interfaces has to incorporate techniques for processing and understanding Natural Languages. However the multitude of different linguistic aspects of Natural Languages, together with the inherent ambiguity to the latter, poses a great challenge to those interested in analysing ordinary language and to represent it in a formal way. A good form of make analyses of Natural Language linguistic aspects is to create a formal representation to isolate the essential ingredients of the phenomenon of interest in a somewhat precise and structure manner.

Drawing on some of the ideas and concepts underlying H.Kamp’s Discourse Representation Theory, but adopting a more proof-theoretic perspective, this paper proposes a representation system for Natural Language based on D. Gabbay’s Labelled Deductive Systems.

Key Words: Discourse Representation, Natural Language, Labelled Deductive System.

1 Introduction

The use of natural languages is very different from the use of formal languages such as computer languages or logic languages. Languages are used to represent information as well as to communicate this information. Languages can be used for communication between two or more people, people and machine or two or more machines. The difference between natural languages and formal languages is that natural languages have many ambiguities. People communicate in a natural language in many different ways. This fact creates a great number of different characteristics in natural languages that do not exist in formal languages. This feature makes NL difficult to be analysed and comprehensible in formal ways.

The representation of NL discourse fragments in a formal language is used to analyse some characteristics that create ambiguity. The discourse information is represented using the formal language. Some of the analysed phenomena discussed here are pronominal anaphora, universal and existential quantification, conditionals and negation.

Before translating a discourse in NL into a formal representation we need to construct a grammatical structure of the sentences of this discourse. This structure will define the grammatical functions of all the words of the sentences. To construct such a structure we use a grammar. The grammar is formed by a set of constructive rules and insertion rules. The constructive rules define the structure of the sentences and the insertion rules define the function of each word. NL have too many different aspects and linguistic features. It is difficult to define which aspects are important to be analysed and which are not. Moreover, we can analyse an infinity number of different linguistic aspects. We define a fragment of the language whith some chosen aspects. This fragment is defined by the insertion grammar rules.
All the existing NL have ambiguities. Some of the interesting linguistic aspects occur in many of this languages, while some are features of only a few studied languages. We choose to study the English language, since English is much more studied than any other language. The proposed representation system can be used to virtually any NL with some modifications. Nevertheless the modifications will be needed to adapt the formal representation to the different aspects of the different languages.

This representation system is based on LDS (Labelled Deductive Systems), a natural deduction system for logic languages, proposed by Dov Gabbay [Gab91]. This system has been adapted to represent the aspects of the English fragment defined here. (Other aspects of English can also be taken to extend the fragment under study and make the system more complete.

Discourse in NL refers to objects and facts of some reality. Such a reality can be the known world or some specific reality. The discourses qualify this objects. With LDS it is possible to represent all the referents that point to objects referred to in by the discourse, the predicates that qualify those referents and indicate from where referents and predicates come from. We can manipulate the different realities where the referents exist, and take into account the ambiguities created by this realities.

Various representation systems for NL have been proposed in the literature. A good example of this kind of system is the (DRT) Discourse Representation Theory [KR93].

The intention is to build a deductive system that can be used as the basis for software systems for natural language processing, supporting intelligent interfaces.

2 LDS

For those not familiar with the LDS perspective, it suffices at this stage to say that the declarative unit of logical systems is seen as made up of two components: a formula and a label. The label is meant to carry information which may be of a less declarative nature than that carried by the formulas. The introduction of such an ‘extra’ dimension was motivated by the need to cope with the demands of computer science applications.

Indeed, with the diversification of computer science applications to problems involving reasoning, there has been a proliferation of logics originated mainly from the need to tailor the logical system to the demands of the particular application area. If there were a number of ‘logics’ already developed and well established in the mathematical and philosophical logic literature (relevant, intuitionistic, minimal, etc.), the diversification was significantly increased with the contribution from computer science.

Gabbay observed that many of the distinctive features of most logics being studied by logicians and computer scientists alike, stemmed from ‘meta-level’ considerations: in order to consider a step to be a valid one, it was invariably the case that one had to take into account questions like: ‘whether the assumptions have actually been used’; etc.

There are a number of inconveniences in having to cope with increasingly diverse logical systems, and Gabbay set out a research programme with at least the following desiderata: (i) to find a unifying framework (sequent calculus by itself would not do, and we shall see why later on) factoring out meta- from object- level features; (ii) to keep the logic (and logical steps, for that matter) simple, handling meta-level features via a separate, yet harmonious calculus; (iii) to have means of structuring and combining logics; (iv) to make sure the relevant assumptions in a deduction are uncovered, paying more attention to the explicitation and use of resources.

The idea of labelled deduction seemed to be a natural evolution from the traditional logical systems. The development of a novel approach to logic, namely Labelled Deductive Systems, where the meta-level features would be incorporated into the deductive calculus in an orderly manner, looked general enough to be an
appropriate candidate for such an unifying framework. Indeed, it seems fair to say that Labelled Deductive Systems offer a new perspective on the discipline of logic and computation. Arising from computer science applications, it provides the essential ingredients for a framework whereby one can study: (i) meta-level features of logical systems, by 'knocking down' some of the elements of the meta-level reasoning to the object-level, and allowing each logical step to take care of what has been done so far; (ii) the 'logic' of Skolem functions and substitution (dependencies, term declaration).

The functional interpretation of logical connectives is concerned with a certain harmony between, on the one hand, a functional calculus on the expressions built up from the recording of the deduction steps (the labels), and, on the other hand, a logical calculus on the formulas. It has been associated with Curry's early discovery of the correspondence between the axioms of intuitionistic implicational logic and the type schemes of the so-called 'combinators' of Combinatory Logic [Cur34], and has been referred to as the formulae-as-types interpretation. Howard's [How80] extension of the formulae-as-types paradigm to full intuitionistic first order predicate logic meant that the interpretation has since been referred to as the 'Curry-Howard' functional interpretation. Although Heyting's [Hey56] intuitionistic logic did fit well into the formulae-as-types paradigm, it seems fair to say that, since Tait's [Tai65] intensional interpretations of Gödel's [Göd58] Dialectica system of functionals of finite type, there has been enough indication that the framework would also be applicable to logics beyond the realm of intuitionism. Ultimately, the origins and the foundations of a functional approach to formal logic are to be found in Frege's system of 'concept writing', not in Curry, or Howard, or indeed Heyting. In this perspective, we shall be working on a labelled natural deduction system [dQG92] which we would like to see as a reinterpretation of Frege's 'functional' account of logic: it is as if the theory of functions of Grundgesetze is put together with the theory of predicates of Begriffsschrift.

The LDS perspective provides a convenient way of modelling natural language phenomena incorporating the underlying device of keeping track of proof steps, thus accounting for dependencies.

The rules for introduction and elimination of a labelled natural deduction are, for example:

\[
\begin{align*}
\frac{a:A \quad b:B}{(a,b):A \land B} & \quad \text{\textit{\textbf{\text{-l}}}} \\
\frac{FSI(a):A}{(a,b):A \land B} & \quad \text{\textit{\textbf{\text{E1}}}} \\
\frac{SNI(b):B}{(a,b):A \land B} & \quad \text{\textit{\textbf{\text{E2}}}} \\
\frac{x:A}{\text{inl}((a, x)):A \lor B} & \quad \text{\textit{\textbf{\text{-l1}}}} \\
\frac{y:B}{\text{inr}((b, y)):A \lor B} & \quad \text{\textit{\textbf{\text{-l2}}}} \\
\frac{[a:A] \quad [b:B]}{\text{CASE}((d, a, f(a), b, f(b)):C}} & \quad \text{\textit{\textbf{\text{-E}}} }
\end{align*}
\]

\[
\frac{\lambda x. y:A \rightarrow B}{\text{APP}(y, x):B} \quad \text{\textit{\textbf{\text{-E}}}}
\]

3 DRT

Discourse Representation Theory constructs a semantic representation of sentences in a discourse. This semantic representation is called Discourse Representation Structure (DRS). The DRS's can represent from single sentences to complete discourse fragments or texts. The structures are used to construct an analysis of many different features of the language.

A DRS consists of two components: a set of discourse referents and a set of conditions. The referents represent the objects indicated by the subject and object of the sentences. The conditions are predicates whose arguments are the referents. So there are objects and conditions that qualify and relate those objects.

The DRS's use letters as referents, and conditions to define the referent's type and to relate the referents (Figure 1).

The components of the DRS are grouped in a box. Some conditions are constructed as new DRS's, inside the main DRS. The main DRS represents the discourse been analysed. The construction of this representation
structure is recursive. Moreover, boxes serve to delimit scope and accessibility of referents with respect to other boxes.

4 Grammar

Before using the system to represent the discourse, we need to construct a syntactic analysis of the sentences. This can be made using grammar rules to construct a structural tree that represents the syntactic functions of the words that compound the sentences.

A grammar is a set of rules that specify the sentences that can be constructed in a language. There are many different sort of grammar to different uses. We use a formal grammar whose rules help us construct derivation trees (or grammar trees) to represent the syntactical function of the words and their relations. The tree shows the role of the words in the sentence.

In a grammar we have two kinds of rules: the construction rules and the insertion rules. The first are used to construct the tree and the second to insert the words in this structure. The grammar proposed here define the English fragment studied.

Rules

\[
\begin{align*}
S & \rightarrow \text{NP} \ \text{VP}' \ | \ \text{COND} \\
\text{NP} & \rightarrow \text{PN} \ | \ \text{PRON} \ | \ \emptyset \ | \ \text{DET} \ \text{N} \\
\text{VP}' & \rightarrow \text{AUX} \ \text{not} \ \text{VP} \ | \ \text{VP} \\
\text{VP} & \rightarrow \text{V} \ \text{NP} \ | \ \text{V} \\
\text{N} & \rightarrow \text{N} \ | \ \text{RC} \ | \ \emptyset \\
\text{RC} & \rightarrow \text{RPRO} \ \text{S} \\
\text{COND} & \rightarrow \text{if} \ S \ \text{then} \ S \ | \ \text{suppose} \ S \ \text{then} \ S \\
\text{DET} & \rightarrow \text{a} \ | \ \text{every} \ | \ \text{the} \ | \ \text{some} \\
\text{PRON} & \rightarrow \text{he} \ | \ \text{him} \ | \ \text{she} \ | \ \text{her} \ | \ \text{it} \ | \ \text{I} \ | \ \text{me} \ | \ \text{we} \ | \ \text{us} \ | \ \text{they} \ | \ \text{them} \ | \ \text{you} \\
\text{PN} & \rightarrow \text{John} \ | \ \text{Mary} \ | \ \text{Peter} \ | \ \text{Elizabeth} \ | \ \text{Ann} \ | \ \text{Tractatus} \ | \ \text{The Tao of Physics} \ | \ \text{New York} \\
\text{N} & \rightarrow \text{person} \ | \ \text{man} \ | \ \text{woman} \ | \ \text{book} \ | \ \text{car} \ | \ \text{dog} \ | \ \text{cat} \ | \ \text{horse} \ | \ \text{Porsche} \\
\text{AUX} & \rightarrow \text{does} \ | \ \text{do} \ | \ \text{did} \ | \ \text{will} \\
\text{V} & \rightarrow \text{like} \ | \ \text{own} \ | \ \text{love} \ | \ \text{drive} \ | \ \text{kiss} \\
\text{RPRO} & \rightarrow \text{who} \ | \ \text{which} \ | \ \text{that} \ | \ \text{whom}
\end{align*}
\]

The abridged names used have the following meaning: \( S \) - \text{SENTENCE}, \( \text{NP} \) - \text{NOUN PHRASE}, \( \text{VP} \) -
Noun phrases are related to objects, so they are used to generate referents. Verbs represent relations between referents, and are represented by predicates. Every single sentence can be represented by a predicate, generated by the verb. The predicate will qualify the relation between the subject and the object.

To construct the sentences of the language fragment defined, we have to consider agreement rules like number, case, gender and others. This rules define situations where the words can be used or not. They also help find references between pronouns and referents. In a general way the agreement aspects are: Number, Gender, Case, Verbal Transitivity and Verbal Form. These aspects modify the grammar rules in a way that it is possible to define the correct pronoun or name or verbal form to use.

The referents defined by proper names or indefinite descriptions are qualified in some way. This qualification is transformed into predicates in the representation system. There is an algorithm to identify this qualification and a table to take the information used by the algorithm.

5 Natural Language

Sentences of a discourse fragment have many informations that have to be represented. This sentences talk about objects of the world and about facts and relations between this objects. A system that is proposed to represent this information need to have diferent structures to each kind of information, and more, need to have a way of take this information structures together. Indeed more, this system need to be capable of make deductions to construct a global information that is the discourse representation. The representation structures are described in the next sections.

5.1 Simple sentences

Simple sentences with proper names as subject and object, are the simplest kind of sentences to be studied. Proper names reference objects of the world. We take proper names as referents to those objects. Before constructing a representation, a grammatical analysis is made. This analysis construct a derivation tree for the sentence [Figure 2]. A representation structure, a deduction tree, is created for the sentence using the derivation tree.

John loves Mary

\[
\text{(text(John); Person(John); text(Mary); Person(Mary))}\quad t:\text{Loves(John, Mary)}
\]

The way to construct the representation structure have been organized in rules and algorithms.

Algorithm - PN (Proper Name):

- Create a referent for the proper name, using the name itself, and construct an expression for this referent in the form \text{text}(x) : P(x) where \text{text} indicates from where the referent came, \(x\) is the referent and \(P\) is the predicate that qualifies the referent.

Algorithm - ID (Indefinite Description):
Figure 2: Derivation Tree for grammatical analysis

- Create a referent for the indefinite description using a letter as referent, and construct an expression for this referent in the form text(x) : P(x) where text indicates from where the referent came, x is the referent and P is the predicate that qualifies the referent.

**Introduction Rule for Conjunction:** this rule is the same as the introduction rule for conjunction, defined for LDS.

The next algorithm describes how a structure representation for a simple sentence can be constructed.

**Algorithm - SS**

- Create the referents and expressions for this referents for noun phrases using the Algorithm - PN and the Algorithm - ID.
- Make a deduction with the referent expressions using the introduction rule for conjunction.
- Create an expression for the verb with a letter or a word in the label. The formula is of the form P(x, y) where P is the predicate based on the verb and x and y are referents related by the verb in the sentence (subject and object).
- Make a deduction with the last two expressions using the introduction rule for conjunction.

The l.h.s. of the expression, the label, represents information from the meta-level. It shows from where the referents come. It links the referents to the objects in the realities they exist. It shows that the referents were brought from the meta-level to the object-level and that the predicate based on the verb could be constructed on the basis of the existence of the meta-level information. The r.h.s. is formed by formulas, using an object language. The predicates qualify the referents. This object language deals with a first order logic calculus.

The word text in the label indicates that the referent comes from the text been analyzed. If this information had been taken from any other database, it could be in the label substituting the word text for some word indicating the database.
5.2 Anaphora

One problem that appears in any discourse analysis is the resolution of anaphoric references. It happens when a pronoun in a sentence makes reference to an object described in other sentences. The difficulty is to find the referent to which the pronoun makes reference.

John owns a dog. He likes it.

Here there is a conjunction of sentences where the second sentence pronouns make reference to the first sentence names (they are the same objects, the same referents).

The anaphoric reference can be resolved by an algorithm that defines the characteristics of the pronouns like number and gender, and searches for some defined referents with the same characteristics. If there are more than one alternative referents we choose the last defined.

Algorithm - CSS - Conjunction of Simple Sentences:

- Create the representation for the first sentence, as a single sentence.
- Create the representation for the second sentence, as a single sentence. If this sentence has pronouns, resolve the anaphoric relationship, using the existing referents referenced. This referents does not create new expressions.
- Make a deduction with this two sentence representations using the introduction rule for conjunction.

5.3 Relative clause

Another kind of group of sentences similar to the previously studied situation is the case where relative clauses occur. In this case there is a relative pronoun that makes reference to some referent.

John likes Mary whom Peter loves.

The grammatical analysis turns this sentence into a conjunction of two sentences where the object of the first sentence is the same as the object of the second. Then the structure is constructed in the same way used in the last example.
5.4 Negation

When we use negative sentences, another kind of difficulty is involved in the analysis. The use of proper names or indefinite descriptions as the object of the sentence makes a big difference in the interpretation and representation constructed.

John does not like Mary.
John does not own a dog.

These two sentences have negative information, since they say that something is not the case. However, they are different in the way they refer to their objects and construct referents. In the first sentence we have a proper name as object of the negation. The object defined by the proper name exists despite the negation. So the negation is applied only at the predicate defined by the verb.

In the second sentence, the object is an indefinite description, which does not define a specific existing object. The negation says that there is no dog such that John owns it.

Thus when the object is a proper name, it is a referent and has its own force. In this case the referent is out of the effect of negation. In the case of an indefinite description, the possible object is qualified but is not defined, so we say that there is no object that satisfies the predicates.

The representations created for this two kind of sentences are different too.

John does not like Mary.

$$\text{John does not like Mary.}$$

$$\text{Like}(\text{John}, \text{Mary})$$

John does not own a dog.

$$\text{John does not own a dog.}$$

$$\text{Own}(\text{John}, x)$$

When the negative sentence has pronouns as subject or object, it has to be resolved as pronoun anaphora. The resolution is the same as used to Conjunction of Simple Sentences.

John owns a dog. He does not like it.

$$\text{John owns a dog. He does not like it.}$$

$$\text{Own}(\text{John}, x)$$

The second sentence makes reference to existing objects, defined in the first sentence, namely 'John and a dog'.

When the first sentence is negative and the second sentence has pronouns additional problems may occur.
John does not own the Tractatus. He likes it.

In this case it is possible to construct a link between the pronoun it and a referent, because this referent exists. The fact is that there is an object that is a book, called Tractatus and that John does not own this object.

**Introduction Rule for Negation:**

$$\frac{x:A}{\lambda x. f(x): \neg A}$$

**Algorithm - Negation:**

- If the subject or the object are proper names create referents for them and expressions for these referents.
- If the subject or the object is a pronoun then resolve the anaphoric relation.
- If there is more than one referent expression make a deduction with them using the introduction rule for conjunction.
- If the object is a proper name or a pronoun then
  - Create an expression for the verb where the label is a letter and the formula is of the form $P(x, y)$. In this expression $P$ is the predicate based on the verb and $x$ and $y$ are the referents related by the verb.
  - Make a deduction with the verb expression using the Introduction Rule for Negation. The label of the false expression is of the form $f(o)$ where $o$ is the label of the verb expression.
  - Make a deduction with the referents expression (if it exists) and the negated expression using the Introduction Rule for Conjunction.
- Else if the object is an indefinity description then
  - Create an expression for the suposed object and the verb where the label is a letter and the formula is of the form $\exists x(a)P(y, x)$. In this formula $x$ is the indefinity description, $a$ is its type, $P$ is the predicate based on the verb and $y$ is the referent created by the subject.
  - Make a deduction with this expression using the Introduction Rule for Negation. The label of this expression is of the form $f(o, x)$ where $o$ is the label of the verb expression and $x$ is the suposed referent.
  - If the subject referent expression exists make a deduction with the referent expression and the negated expression using the Introduction Rule for Negation.

**Final remarks**

In the current stage of the project, the system can analyse a small fragment of the language under study. Our future aim is to extend this fragment, with a view to be able to analyse more linguistic phenomena as well as to construct a better analyses of the language.
The advantage of this system is its simplicity and the possibility to represent different aspects and phenomena existing in the discourse in separating ways: the referents qualifications and relations in the formulas, and a pointer to the referents and an information of how the referents and relations could be constructed.

References


