NATURAL LANGUAGE COMPUTATION MACHINERY

Helder Coelho
Laboratório Nacional de Engenharia Civil
Lisboa, Portugal

Resúmen

Along natural language (NL) conversations (multi-agent interactions) it is rather important to have access to several learning apparatus in order to maintain adaptation flexibility among knowledge-based systems and between any of these systems and some human being.

From early 80's my group at LNEC has been interested in understanding discourse at large and in building up a theory of conversations. Recently, we realized a series of experiments concerning learning to compute unknown features, such as new words and their syntactic categories, new gramatical concepts and new syntactic rules. The major goal was the improvemento of NL sensitivity to error of an intelligent tutor for logic programming and Prolog language, but diverse side-effects could be taken into account, namely the generalisation of these ideas to enhance also dialogue grammars. In point of fact, behind this research there is a general thesis that consists of proving that grammatical knowledge can never be complete in an open world and, therefore, any grammar of communication must always evolve.

"The most-sophisticated existing computer systems
have no flexibility or adaptability or
tolerance for error"

Douglas Hofstadter

1972, SHRDLU demonstrated that

a computer could carry on
a simple conversation.

Natural Language Understanding
research led to the fundamental problems of

REPRESENTATION and REASONING.

"The early optimism was unrealistic".

T. Winograd, 1987

The old question continue to challenge us,
and new kinds of questions have arisen.

Researching shifted their focus to
higher level structures of language
such as **conversation theory**,
which could be applied to designing
coordination systems that do not require
reasoning techniques that mirror common sense.

CURRENT APPLICATIONS OF NL PROCESSING SYSTEMS

- intelligent database retrieval
- text skimming and summarization
- machine translation
- user interfaces to expert systems

LIMITATIONS OF ONGOING SYSTEMS

- handling of single sentences with any generality

THE **FULL POTENTIAL** OF NL INTERFACES HAS TO BE REALIZED!

- WHICH **TECHNIQUES** ARE NECESSARY FOR THE **NEXT GENERATION** OF NL SYSTEMS THAT CAN HANDLE MULTI-SENTENCES TEXTS AND INTERACTIONS?

(AAAI-88 Tutorial MA5 "Natural language — Beyond Single Sentences Systems"
James F. Allen and Bonnie L. Webber)

Single - Sentence ⟶ Multi - Sentence

database query systems | user interfaces to expert systems

⇒ How can we interpret **text** and **extended dialog** in context ?

Main topics for James F. Allen and Bonnie L. Webber (1988)

Incremental modeling - for mapping between linear texts and complex representations of meaning

Discourse devices - for reducing the processing load on the system

Planning and plan recognition - for facilitating cooperative interaction

PSYCHOLOGICALLY DRIVEN MODEL

partial interpretations of sentences are simple processors that fight among themselves for superiority.

Understanding of language = collection of coperating partial interpretations that survive.

FOCUS-OF-ATTENTION MECHANISM

A capability should exist for assignin
a context in which certain rules apply
There should also be a capability fc
focusing the system's resources whe
a significant event occurs.

AI PROBLEMS:

are problems **shared** with most or all
of the mental and social disciplines
of psychology, philosophy, economics,
politics, sociology and management.

eg.

RATIONALITY,
MOTIVATION,
GROWTH (Adaptation, Improvement, Learning),
PURPOSE (Meaning, Design),
PERCEPTION,
REPRESENTATION (Imagination), and
COMMUNICATION.

MIND = Structure cruncher or Statistic cruncher ?

A **mind** is a statistical sensitive engine processing numerical representations that are structure-sensitive.

RESEARCH QUESTIONS

Q1 : Can we achieve transportability of NL interfaces? Are there universal interfaces?

Q2 : Is complete understanding necessary?

Q3 : How can we cope with noisy understanding?

Q4 : Are open worlds really open?

Q5 : Can situations induce specific attitudes?

TOTORIAL DIALOGUES are

a special case of CONVERSATIONS in general.

and

INTELLIGENT TUTORING SYSTEMS = KNOWLEDGE COMMUNICATION SYSTEMS

(3 components)

- **the knowledge to be communicated**
- **the student´s knowledge at anytime (the student model)**
- **the knowledge of how to teach or communicate that knowledge**

+

information about the learners´s interests, aptitudes, motivations, recent activities

⇓

what the student <u>knows</u> (KB) **versus** **what the student believes** (BB)

EXPERIMENT1 (LNEC)

ANY SYSTEM (eg. Intelligent Tutor System) to be **flexible** needs:

- **dynamic and modular architecture**

  to be automatically reconfigured during any working session

- **dynamic evolving capability**

  by learning techniques

- **to follow and guide user's reasoning processes**

  during problem solving

- **to adapt to each user**

  including its manner of expression in a NL

Tutor's learning of NL knowledge relies on student information and serves its capability to improve student guidance.

Guidance activity is accomplished through the use of:

- menus
- written NL
- graphical facilities

SYNTACTIC DESCRIPTION OF NL

a ⟶ b,c — context-free grammar rule

syntactic categories

A ⟶ B,C — logic grammar rule $\left\{\begin{array}{l}\text{DCG}\\\text{XG}\\\text{GG}\\\text{RLG}\end{array}\right.$

meta-variables denoting relations between syntactic categories of a, b and c word strings having these categories

A ⟶ [x] .

word categorized as an a

meta-variable denoting a relation among x, a and other information.

A ⟶ {D1}, B, {D2}, C, {D3}.

A ⟶ {D4}, [x], {5}.

D1, D2, D3, D4 and D5 are control descriptions

Therefore:

* capability for automatic **syntactic** guided spelling correction

learning new ...

words | grammatical concepts | syntactic rules

* capability for automatic **semantic** guidance

learning new ...

semantic entities

* capability for automatic **pragmatic** guidance

learning new ...

attitudes | dialogues modes (grammar rules) | situations (contexts)

Principles for writing grammar rules:

(1) the meta-interpretation process is made easier because one can choose **to focus attention** on B or on C, if it is necessary to prove a string of words as belonging to category A.

(2) at the lexical level, the category of each word defines the syntactic context where it may appear, the categories of its pre and post modifiers.

Hypothesis for execution failure of a grammar rule:

1) First word of the NL input string is unknown because it is **misspelled**,

2) First word of the NL input string is **unknown**,

3) First word of the NL input string is **known**, but its category does not match the category denoted or required by B.

LEARNING TO COMPUTE UNKNOWN FEATURES

— enhancement of mixed initiatives

Learning
- single activities
- sequence activities
  - context units
  - plan units

LEARNING UNKNOWN RULES AND CATEGORIES

A ––> B , C .	(1)

A ––> B1 , C .	(1′)
B1 ––> New , B .	B1 is higher level category than B
New ––> [ ] .	

or

New ––> [Wi] .

PROCEDURE FOR GRAMMAR RULE REWRITING:

- Determine conflict category
  (it was B in above explanation).

- Generate and test hypothesis for rewriting conflict category.

- Rename conflict category (B1).

- For each grammar rule, where the conflict category appears in its body (the right hand side of the focussed rule), replace conflict category by its new name, forget the old grammar rule and add the new corresponding rule.

- For each modified grammar rule, add to the pupil´s model two reminders for future action: one for cleaning from the tutor´s model the old rule that has been modified; the other for adding the newly rewritten rule.

- Create a new rule having the renamed conflict category as it head and, as its body, the word (or string of words) category, confirmed by the user, followed by old conflict category name (this is related with creation of rule (4), in the above explanation).

- Add to the pupil´s model a reminder for future action, in order to add this new rule to working memory.

- Create a rule having as its head the prefix category found for the conflict category, and as its head, and an empty list as its body, is necessary for handling cases where the conflict category is not prefixed.

- Add to the pupil´s model a reminder for future action, in order to add to working memory the lastest newly created rule(s), for when this model will next be involved in a teaching session.

This is a very simple context-free grammar for Portuguese

```
s --> np, vp                        (R1)
np --> det, np_nucleus              (R2)
np --> pronoun                      (R3)
np_nucleus --> proper_noun          (R4)
np_nucleus --> noun, adj            (R5)
vp --> vp_nucleus, pp               (R6)
vp_nucleus --> be, be_comp          (R7)
vp_nucleus --> vi                   (R8)
vp_nucleus --> vt, np               (R9)
be_comp --> adj                     (R10)
be_comp --> np                      (R11)
pp --> prep, np                     (R12)
pp --> []                           (R13)
adj --> []                          (R14)
```

=> Que são aquelas bonitas flores? | => What are those beautifull flowers?

Na sua frase, **bonitas** é um adjectivo. Não é verdade? (**Sim** / não) | In your sentence, **beautiful** is an adjective Isn't it? (**Yes** / No)

Na sua frase, **flores** é um substantivo. | In your sentence, **flowers** is a noun.

Não é verdade? (**Sim** / não) | Isn't it? (**Yes** /No)

Double confirmation generates the hypothesis that a new kind of noun description, **noun1**, must be added

noun1 --> adj, noun (R15)

and, because rule (R5) is the only one where old **noun** category appears at its right side, it must be substituted by

np_nucleus --> noun1, adj (R5')

Consider that the tutor does not know **adverbs**.

=> As rosas são flores **muito** bonitas. | => Roses are **very** beautiful flowers.

Não conheço a palavra **muito**. | I don't know the word **very**.

Qual é a sua categoria sintáctica? | Which is its syntactic category?

CATEGORIAS	**CATEGORIES**
determinante	determiner
substantivo	noun
pronome	pronoun
adjectivo	adjective
verbo	verb
outra	**another**

=> É um advérbio. | => it's an adverb.

As a consequence one must complete the pupil´s model with rules

```
adj1 --> adverb, adj              (R16)

adverb --> []                     (R17)

adverb --> [muito]                (R18)
```

substitute rules (R5´), (R10) and (R15) by the new rules

```
be_comp --> adj1                  (R10´)

np_nucleus --> noun1, adj1        (R5´´)

noun1 --> adj1, noun              (R15´)
```

EXPERIMENT 2 (University of Birmingham)

APPLICABILITY OF DCG's : FORMS OF ASSISTANCE

goals	- Input prediction - Error recognition - Explanation	Conditions	no domain state information no user modelling

AI techniques mostly used: DCG's
planning
default reasoning

Tool : Prolog

Application area : Intelligent Tutoring Systems

Recognise user plans by examination of the dialogue between the user and the interactive application for which tuition is to be offered.

method:
- the user input is perturbed to try and produce sentences that can be parsed against a given user plan definition.
- the program provides the repaired user plan and a text to indicate the type of bug.

difficulty:
- filter perturbed (noisy) inputs

Interpretations of possible **perturbed versions** can be grouped into:

A) **Recognised**

B) **Single buggy terminal symbol**

— incorrect terminal symbol

— missing terminal symbol

— extra terminal symbol

C) **User misconception**

— incorrect understanding of a user plan

— better way of achieving a desired end result

— incomplete user plan

D) **Incomplete user plan**

Technical aspects

- the head of each definite clause for a user plan definition has a single argument that specifies the user plan definition name, the composition of the syntaxe tree and a specification of the processes involved and the inputs and outputs to the processes.

- the user plan definition name is present as the argument's functor as opposed to being the definite clause's predicate symbol (ie. the name of the non-terminal symbol) so that user plan definitions can be accessed as data.

- this feature is attractive for a separately packaged tutor or user plan scheduler.

- the syntax tree is used as an internal representation of the parse.

- the inputs and outputs are present to provide semantic information.

Implementation

Incorrect terminal symbol

Each terminal symbol is replaced by an uninstantiated logic variable and the program attempts to parse each resulting sentence against the given user plan definition.

Missing terminal symbol

A variable terminal symbol in inserted at all possible places in the user input and the program attempts to parse each resulting semtence against the user plan definition.

Extra terminal symbol

Each terminal symbol is removed from the user input and the program attempts to parse each resulting semtence against the given user plan definition.

CONVERSATION CAN BE STUDIED
ACCORDING TO A MODEL
WHERE THE MAIN CONCEPT IS A THEORY
AND WHERE EXCHANGES ARE GOVERNED
BY **SYNTACTICAL, SEMANTICAL** AND
PRAGMATIC DEVICES.

SITUATIONS PLAY A DOMINANT ROLE
AND CONTEXT MARKS ARE MEANINGFUL.

EXPERIMENT 3 (LNEC)

PRAGMATIC ATTACHMENT DEVICES

Main thesis: (ITS)

1) the process of teaching is conducted along different layers of complexity and facing a reactive environment where interactions have a guidance role;

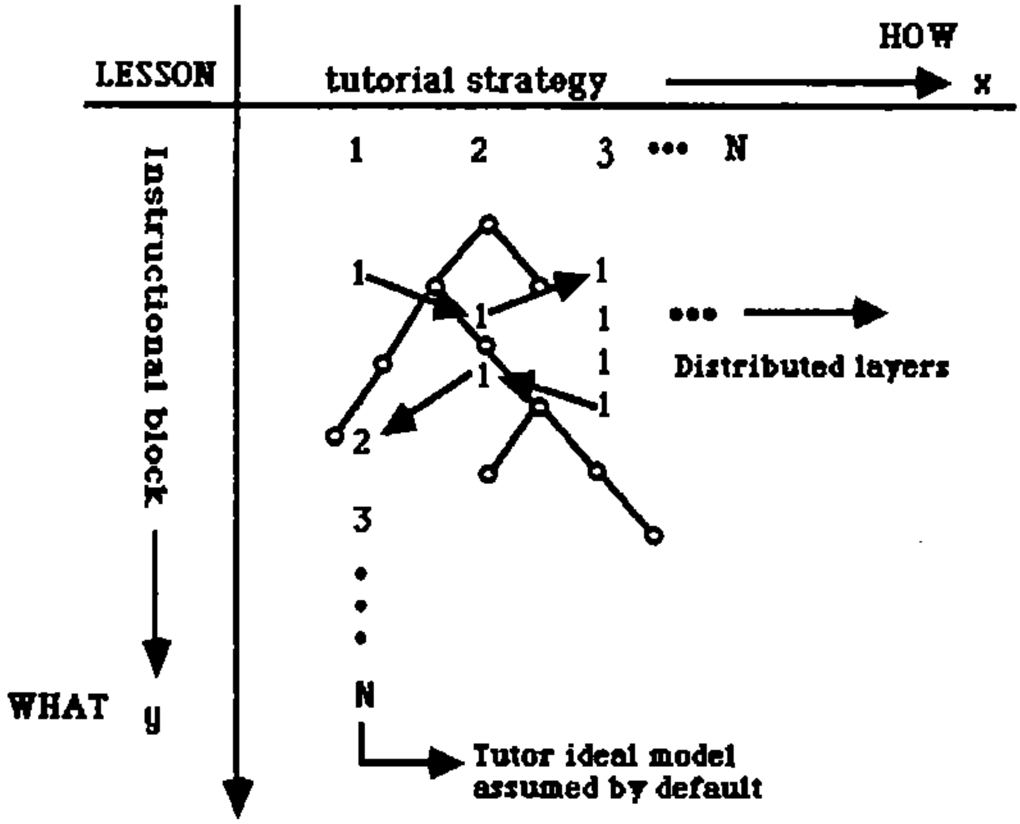
2) the process of apprenticeship is conducted by hypothesis generation;

3) the student profile is obtained simultaneously by monitoring his qualitative evolution in answering and by identification of his misconceptions.

The tutor is able to converse and to teach by **mental image guidance**. It generates an abstracted image by constructing a **model of the student** with several layers of complexity.

STUDENT IDEAL MODEL (MENTAL IMAGE)

2 1/2D : 1 dimension for **how** tutorial strategies act
1 dimension for **what** instructional blocks are
1/2 dimension for layer thickness



student doings in a set of lesson(x,y,h)

lesson content to be presented

description of an instructional path

actual instructional block

- Student past and evolution of his own knowledge
- mostly used for choosing tutorial strategies

PRAGMATIC ATTACHMENT MACHINERY

— belief generation

— knowledge acquisition and transformation

— teaching decision

CONTEXT organized around the interaction content

Who : actor type

What : speech act type

Where : instruction layer

and the semantic interpretation is supported by the use of models, and it depends on the function attached to each act within a certain scenario.

CONTEXT HISTORY grows up along a net triggered by the use of those instructional layers.

- The tutor measures the student´s performance in each instructional block by taking notes along two students tasks: problem solving and writing examples.
- It is sensitivo to the explanation layer that is more suitable to each student.
- The tutor generates a belief (first order formulae) about the student knowledge and his performance:

B (actor , complexity_layer (x))   simple structure

Modal operator capable to represent statement about beliefs

the statement that is believed

the individual that believes (tutor)

B(actor$_1$, B(actor$_2$, <lesson(x,y,h),action_knowledge(K)>)) complex structure

reading: "the actor$_1$ (tutor) believes that the actor$_2$ believes in action_knowledge(K) at lesson(x,y,h)" .

action_knowledge([student_example,error_delection,"message"]) og implementation:

```
hypothesis (tutor, hypothesis (student, p(lesson(goals,2),
                   lesson (goals, 0, yes)),
                   action_knowledge(tutorial_lesson,_,_))).
```

student acquires **knowledge** by observing
student performance in various instructional
oks that a student requires.

is kind of **knowledge** is described by:

$$\left(\bigcup_{1 \le i \le n} \left(B \left(actor_1 , B \left(actor_2 , \langle lesson\ (x,y,h) , action\_knowledge(k)_{[i]} \rangle \right)\right)\right)\right)$$

nion of all the previous beliefs generated during
he set of $\{i\}$ lesson (x,y,h) .

Example:

```
lesson(facts,2,lesson(facts,0,yes)).
lesson(facts,2,lesson(facts,0,no)).
lesson(atom,1,lesson(atom,0,yes)).
lesson(facts,2,lesson(facts,1,yes)).
lesson(facts,2,lesson(facts,2,yes)).
```

The tutor belief, when proved, is transformed in

a statement K about knowledge

statement is represented by a set of formulae

about the student image

used in deciding a teaching strategy or for adequating the tutor model to that student image.

K (actor , complexity_layer (x))

statement that is known

the individual, who has the knowledge

modal operator

Two main transformations:

$$K(actor1, B(actor2, <lesson(x,y,h), action\_knowledge(k)>)) \Rightarrow$$
$$K(actor_1, K(actor_2, <lesson(x,y,h), action\_knowledge(k)>))$$
$$B(actor_1, B(actor_2, <lesson(x,y,h), action\_knowledge(k)>)) \Rightarrow$$
$$K(actor_1, B(actor_2, <lesson(x,y,h), action\_knowledge(k)>))$$

ments B and K may have one of the **four** possible **forms**:

$(actor_1, K(actor_2, <lesson(x,y,h), action\_knowledge(k)>))$
$(actor_1, K(actor_2, <lesson(x,y,h), action\_knowledge(k)>))$
$(actor_1, B(actor_2, <lesson(x,y,h), action\_knowledge(k)>))$
$(actor_1, B(actor_2, <lesson(x,y,h), action\_knowledge(k)>))$

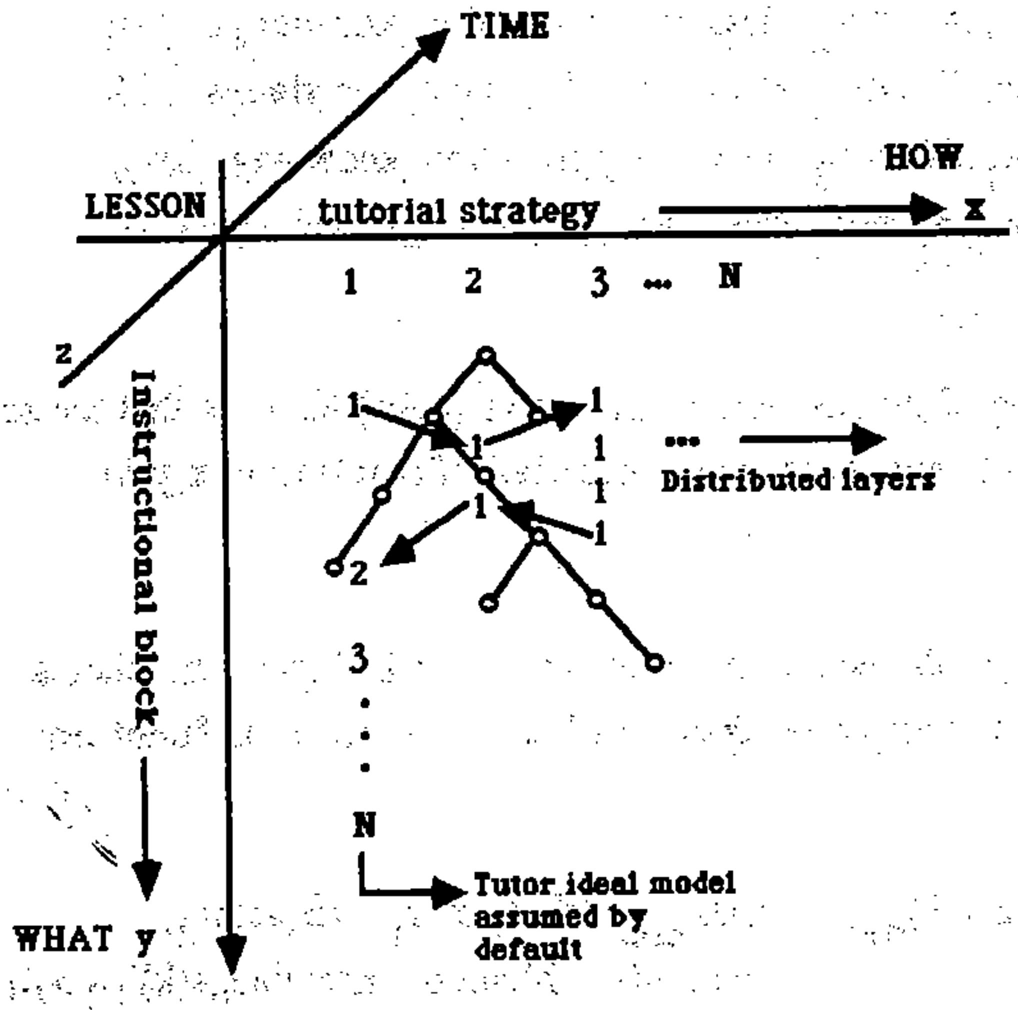
Reading of the first form: "the actor1 (tutor) believes that the actor2 (student) believes in action_knowledge(k) at lesson (x,y,h)".

The tutor **decision to teach** at a certain layer (the choice of a teaching strategy) is taken based upon the following **function**:

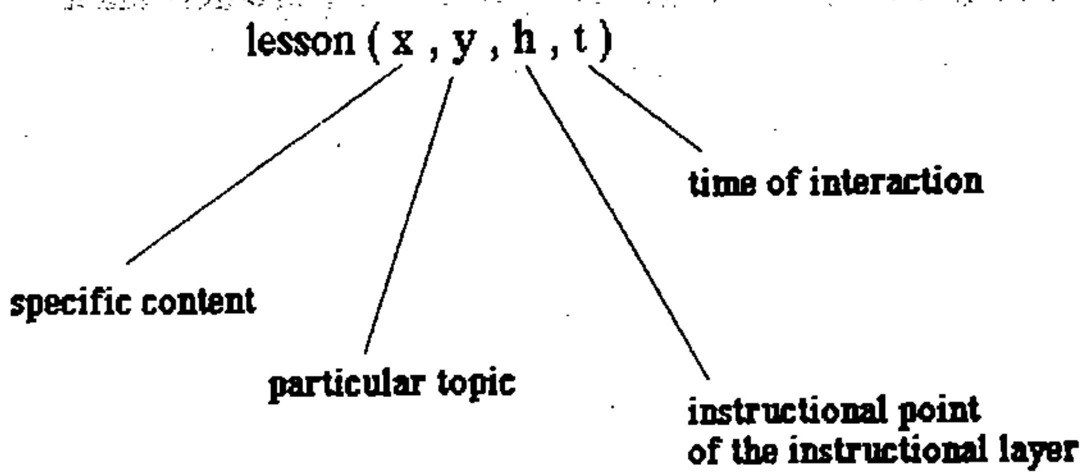
$$lesson(x,y,h)_{n+1} = f(U_{1 \le i \le n}(K/B(actor_1, K/B(actor_2, <lesson(x,y,h), action\_knowledge(k))_{[i]}>)))$$

where (f) is built upon the context history of the conversation taken.

3D TUTOR MODEL



Z dimension for time



MULTI-LAYER LEARNING

What kind of agent is to be blamed for a misunderstanding?

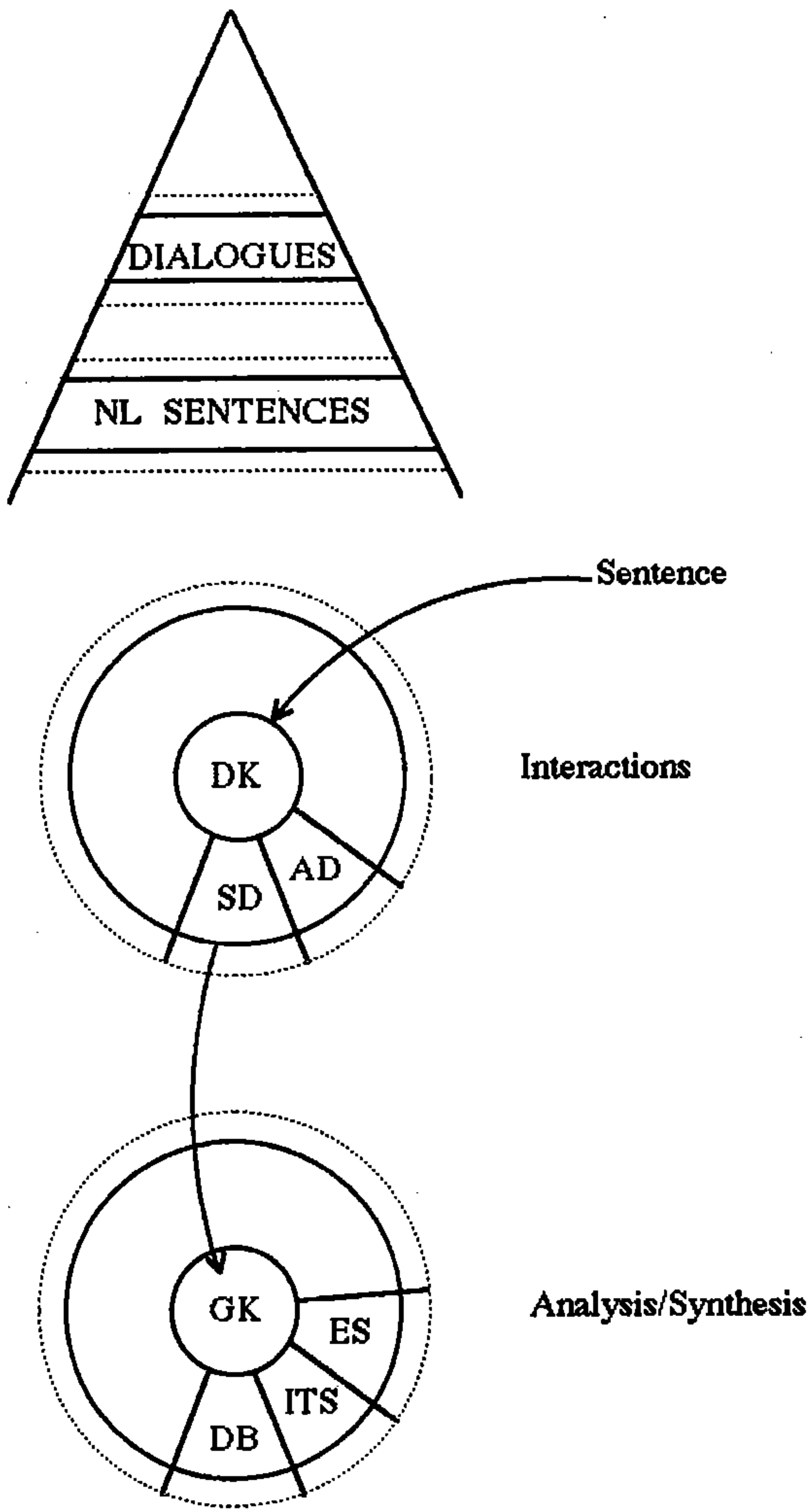
What occurs versus what is expected to occur?

Mathematization of intuition:

$$\text{Error function} = \sum_{n=1} (\text{target}_n - \text{output occurrence}_n)$$

goal: minimize tutor errors

Device: credit assignment for mistakes in order to govern blame!


DIALOGUES
NL SENTENCES
Sentence
DK
AD
SD
Interactions
GK
ES
ITS
DB
Analysis/Synthesis