Reports

of the

Intelligent

Systems

Group

UIUCDCS-F-86-953

CRITICAL ISSUES IN NATURAL LANGUAGE PROCESSING AND THEIR IMPORTANCE TO MACHINE LEARNING

by

Lucja Iwanska

August 1986

ISG 86-10

OF COMPUTER SCIENCE
OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA

File No. UIUCDCS-F-86-953

Critical Issues in Natural Language Processing and their Importance to Machine Learning

Lucja Iwanska*

August 1986

ISG 86-10

Intelligent Systems Group
Department of Computer Science
University of Illinois
at Urbana-Champaign

*On leave from the *Institute of Organization, Ministry of Metallurgy and Mechanical Engineering, Warsaw, Poland*

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I wish to thank Professor Edmund Lipinski from the Warsaw Technical University for turning my attention to natural language processing and Professor Ryszard Michalski for his many valuable comments, criticisms and encouragement. I am also grateful to Professor Gerald DeJong, Mark Goldfain, Peter Haddawy, Bruce Katz, and to Carl Uhrik for useful remarks and proof-reading of the paper.

This work was supported in part by the National Foundation under grant DCR 84-06801, in part by the Office of Naval Research under grant N00014-82-K-0186, and in part by the Defence Advanced Research Projects Agency under grant N00014-K-85-0878.

Critical Issues in Natural Language Processing and their Importance to Machine Learning

Lucja Iwanska

ABSTRACT

Several representative natural language processing systems are reviewed and compared. The review is done from the viewpoint of issues related to the development of systems capable of learning from natural language input. Specifically, such issues are considered as representation of concepts, learning capabilities of the systems, the role of syntax and semantics, restrictions on language and domain, and tradeoff between generality and efficiency. It is shown that further progress in both fields, natural language processing and machine learning, depends on elaborating the theory of concept formation and representation.

1. INTRODUCTION

The idea of communication with machines in natural language came early in the development of computer science, and remains a fascinating research goal today. Out of this idea grew the field of natural language processing (NLP) concerned with the development of the theories and practical implementations of systems for natural language communication. In the course of NLP research a great variety of systems has been implemented, some of them having practical value. Independently, there has been a rapid growth of machine learning, the field concerned with the development of learning systems. This field provides a key to overcoming the bottleneck of knowledge acquisition.

In this context, an exciting research goal is to cross-fertilise NLP and machine learning. Such a cross of the two fields can result in computers that are capable of learning from the information expressed in natural language. This paper addresses selected topics in NLP and machine learning which would bring us closer to such a possibility. The main difficulty which the two fields have to face is the problem of machine representation of concepts that would permit the system to make inferences in an easy and natural way as humans do. The further progress of research in NLP and machine learning depends on the development of a well-grounded concept formation and representation theory.

This paper consists of 5 other sections. **Section 2** discusses two NLP directions: machine translation (Wilks's system, MOPTRANS), natural language interfaces to databases (LUNAR), and one man-machine interaction system- SHRDLU. **Section 3** is devoted to the one of the most controversial problems in NLP, namely, the relation between syntax and semantics. **Section 4** presents the most popular semantic approach to represent knowledge extracted from natural language input- Schank's Conceptual Dependency. **Section 5** discusses three machine learning systems using natural language input: NANOKLAUS, Katz and Winston's system, and GENESIS. **Section 6** compares the analyzed systems and presents criticism. **Section 7** contains suggestions about future research in NLP and machine learning.

2. DIRECTIONS IN NATURAL LANGUAGE PROCESSING

The necessity of learning programming languages discourages most people from widely using computers. Instead of tediously coding the programs one would rather like to ask computers to perform desired tasks in natural language. Within the field of NLP one can distinguish two general directions: machine translation and natural language interfaces; they are discussed in sec. 2.1 and 2.2, respectively.

2.1. MACHINE TRANSLATION

Translation from one language to another was one of the earliest tasks assigned to computers. The initial way to do machine translation was to build a dictionary and grammars for the involved languages, and translation would be a kind of projection from one language to another. The dictionary contained corresponding words and grammatical categories of the two languages, and the grammars specified corresponding structures of the sentences. This literal translation worked relatively well in the case of sentences describing sensory observations or typical concrete activities. For example, the literal translation of the sentences

I see this table.
On Friday I will go to Bari by train.

into German

Ich sehe diesen Tisch.
Am Freitag werde ich mit dem Zug nach Bari fahren.

is fully adequate.

In case of sentences describing mental activities or abstract concepts, literal translation does not work so well. Several classes of sentences for which literal translation fails can be distinguished:

1. **Idiomatic expressions with multiple meanings of words.** In such sentences it is not possible to derive the meaning from the particular words, the expression has a meaning as a whole. In such cases literal translation usually leads to nonsense or a different meaning. Take, for example, the Polish expression *Dziekuje z gory* whose correct translation is *Thank you in advance.* Here *Dziekuje* is *Thank you*, *gora* is a *mountain*, *z* is *from*. Literal translation would give *Thank you from the mountain*, which has clearly a different meaning from the intended one.

   This expression, and similar ones, can, however, be handled by a dictionary look-up. This is possible, because their meaning is not context dependent.

2. **Idiomatic expressions that resulted from some historical events.** A good example for this is the Polish expression *Slowo sie rzeklo, kobylka u plotu.* The literal translation would be *The word was said, a mare is by the fence.* This comes from a historical event in which a Polish King doubting a plan of some nobleman exclaimed: *If you succeed in this I will kiss your mare's ass.* It turned out that the nobleman indeed succeeded in his endevour. So he asked the King to keep his promise, stating that his mare was by the fence. This statement could be translated to *A promise is a promise* [Bulas, Lawrance and Whitfield, 1967]. This translation is, however, rather flat and does not carry the same flavor.

   Expressions in this class can not be handled well by dictionary look-up because simple one-to-one assignment will not preserve the deeper meaning resulting from their historical background. Another reason for the difficulty is that often their proper translation depends on the context in which they appear. For example, the expression *to pull oneself up by one's own bootstraps* has in different contexts two different meanings:

   1. *to succeed on one's own;*
   2. *to convey the feeling that somebody's actions have no chance to succeed; a trial to solve an unsolvable situation;*

3. **Expressions involving words with different scope of meaning.** What Germans mean by *Wald*, French may require *Bois* or *Foret* depending on its size and location. So the German sentence *Ich liebe diesen Wald* would be translated into French *J'aime ce bois*, if one means the grove located in the city, or *J'aime ce foret*, if one means the forest far away from the city.

4. **Sentences with ambiguos words.** Every language has ambiguos words in its vocabulary. English, however, is especially rich on them. Webster's New World Dictionary of the American Language, for example, lists several different entries for most of the words. It is the context which can resolve this type of ambiguity- usually some more or less explicit information from previous sentences. For example, when translating the English sentence *Take away this pipe* we must know the situation behind it in order to decide whether it is a pipe to smoke, a pipe to play, or a pipe for use in construction.

Early machine translation systems that were capable of only literal translation produced low-quality translation, despite much effort and expense. This resulted in pessimism about machine translation, and lead even to the conclusion that the computer translation is not possible. The reason for failure was that in order to produce a good translation a system has to possess knowledge about the world and people, and "understand" the meaning of the words. Some progress in machine translation was made after the development of new knowledge representations based on semantic primitives (see sec.4 about Conceptual Dependency Theory.) The idea was to translate the input text in one language into *pure meaning* that is based on semantic primitives and language independent, and then to translate this internal representation into target language. This approach was implemented in the systems developed by Wilks [1973], Schank [1980] and Lytinen [1984].

2.1.1. WILKS'S SYSTEM

Wilks's system transaltes from English into French, and vice-versa. It splits the input text into fragments and then replaces words in fragments with internal formulae representing the words' meanings and matches the resulting string of formulae against a set of standard forms called templates. These are groups of 3 primitives following the pattern ACTOR-ACTION-OBJECT, for example MAN-CAUSE-MAN. Templates contain mixed syntactic and semantic information. An example of the formula for the verb *drink* is:

```
(( *ANI SUBJ )
 ((( FLOW STUFF ) OBJE )
  (( *ANI IN ) ((( THIS ( *ANI ( THRU PART )) TO ) ( BE CAUSE ))))
```

(*ANI SUBJ) means that action denoted by *drink* is performed by animate subject (ACTOR.)

((FLOW STUFF) OBJE) means that the object of the action is liquid.

((*ANI IN) (((THIS (*ANI (THRU PART)) TO) (BE CAUSE) means that after the action denoted by the verb *drink* liquid is inside the animate subject (ACTOR) and that it was conveyed there through an ACTOR's aperture.

Wilks's system deals with 60 semantic primitives divided into 5 classes. Each class expresses basic entities, states, qualities, and actions about which humans communicate. For example

Class ENTITIES contains primitives like MAN, STUFF etc.
Class ACTIONS includes CAUSE and FLOW primitives.

The result of formulae matching is a first approximation to a semantic representation of each of the fragments. Next, the system ties together these representations to produce a representation for the whole text; it follows the compound patterns that span two templates. For example, the fragments *he ran the mile* and *in four minutes* would be tied together by a compound template TIMELOCATION. The output is generated by unwinding this representation using a function that interprets it in the target language. The system's dictionary contains formulae for all the word senses paired with stereotypes for producing the translated words in the target language. For example, two stereotypes for the verb *advise* (English-French dictionary) are:

```
( ADVICE  ( CONSEILLER A  ( FN1  FOLK MAN ))
          ( CONSEILLER   ( FN2  ACT STATE STUFF )))
```

Functions F1 and F2 distinguish the two possible constructions in French: *conseiller a...* (advice somebody) and *conseiller* (advice something.) Such functions are evaluated by the generation routine.

Wilks's system is able to handle some words and prepositional ambiguity, simple problems of pronoun references, but only within boundaries of a single sentence. Mixing of syntactic and semantic information in templates resultes in duplication of a single syntactic rule in different templates. For example, information that the subject of a verb comes before the verb in English is implicitly encoded into every template that has an action as its second argument. Wilks's semantic based representation of the meaning was developed for the purpose of machine translation and is not good for the other NLP tasks.

2.1.2. MOPTRANS

MOPTRANS (Memory Organisation Packet TRANSlator) developed by Lytinen [1984] is a multi-lingual (Chinese, English, French, German, Spanish) integrated parser which is used for translating short 1-3 sentence newspaper articles about terrorism and crime. It proved that power of the parser can be gained by organising syntax and semantics as two largely separate bodies of knowledge. It allows one to share it across languages. Communication between syntax and semantics is high. The parser builds only a limited amount of syntactic representation during text understanding. Semantics guides the parsing process. The parser produces language-independent, conceptual representations for the stories. It performs frame selection for the stories involving very vague words or phrases, using 6 general purely semantic concept refinment rules operating on the hierarchy of knowledge. Knowledge is organized in a hierarchical manner by using IS-A pointers which point from a structure to more abstract structures, for example, the structure SHOOT points to a more abstract structure HARM.

The languages share as much knowledge as possible. Commonalities in syntactic constructions among the languages such as the fact that English and most romance languages are SVO languages, are reflected in the use of some of the same syntactic rules. Characteristics of such rules are shown below.

Total Number of Syntactic Rules is 285.

Number of Languages Rules are Applicable to	Number of Rules
1	161
2	42
3	54
4	24
5	4

Encoding of words' disambiguation knowledge in terms of *deeper* semantic information resulted in drastically fewer rules for translation of ambiguous or vague words.

The lexically-based approach to syntactic knowledge is incompatible with the task of learning natural language. Lytinen elaborated a set of rules which apply to some words classes instead of particular words, so in case of learning a new word, as soon as one is able to find out its class membership one has all the knowledge necessary to handle it in the text.

Knowledge in the system is handcrafted and extremely difficult to modify. Intuitively, different languages (different cultures) have different concept hierarchies which overlap partially only. Lytinen makes an assumption that these hierarchies are the same. Although MOPTRANS has some important advantages over previous parsers, the quality of its translation still leaves much to be desired. It is meaning preserving but sounds sometimes awkward to native speakers- one has to make some additional inferences in order to understand it. Consider the following examples taken from Lytinen's thesis [1984]:

French input:
Les ambulances de la Croix Rouge ont transporte d'urgence deux jeunes filles, dont les mains avaient ete blessees par suite d'une bombe, a l'hopital Manolo Morales .

English translation:
2 young women who were injured by a bomb in the hands were rushed by an ambulance owned by the Red Cross to the hospital.

German translation:
2 junge Frauen wurden nach das Spital mit einem Krankenwagen von dem Rotkreutz, gehastet. Sie wurden mit einer Bombe verwundet.

A real good translation should be more like:

English translation:
Red Cross ambulances rushed two young women whose hands had been injured as the result of a bomb explosion to Manolo Morales hospital.

German translation:
Zwei junge Frauen, die durch eine Bombenexplosion verwundet worden waren, wurden von einem Rot-Kreutz Rettungswagen mit Blaulicht zum Krankenhaus gefahren.

So far, fully automatic high-quality machine translation remains an unsolved problem. There are programs performing simple phrase-by-phrase translation which is subsequently checked by a human editor. They operate in severly restricted domains only or require large amounts of human post-editing. Some problems as, for example, handling the words with different scope of meaning or disambiguation of words using previous context, are yet to be solved. Machine translation is an active research field of artificial intelligence.

2.2. NATURAL LANGUAGE INTERFACES (LUNAR)

Another direction of NLP is domain-oriented natural language interfaces which enable natural language input or query to programs and systems. They translate natural language input into some formal representation (query languages), perform some processing, such as search in the data base, and return the answer in simple English. Because it is impossible to encompass the entire language, it becomes necessary to constrain it. The question, then, is how much to constrain the language to use it without the significant effort of remembering all the constraints. Such systems were developed for real-world situations, for example, the ticket reservation systems like GUS developed at Xerox Palo Alto Research Center, or Wood's LUNAR system for retrieving information about moon rocks [Woods, 1973]. LUNAR translates questions entered in English into expressions in a formal query language based on the predicate calculus. Parsing is made by an ATN parser coupled with a rule-driven semantic interpretation procedure. The request then is answered in simple English. An example of a request is *What is the average concentration of aluminium in high alkali rocks ?*. Requests are processed in 4 steps:

1. Syntactic analysis using ATN parser and heuristic information (including semantics) to produce the most likely derivation tree for the request;
2. Semantic interpretation to produce a representation of the meaning of the request in a formal query language;
3. Execution of the query language expression on the database;
4. Generating an answer to the request.

LUNAR uses the notion of procedural semantics in which queries were converted into a program to be executed by the information retrieval component.

Ticket reservation systems and LUNAR work well because for such narrowed domains it was possible to predict nearly all the questions and their structures (LUNAR containes a few hundred ATN-networks, the dictionary has 3500 words) and to encode them into the program; quasi-natural language for these systems is very close to a natural one. The cost and effort invested in completing such a hard job were justified because of the large potential user community.

2.3. MAN-MACHINE INTERACTION SYSTEMS (SHRDLU)

We would like future computers to be knowledgeable assistants which would help us with decision-making. They should give intelligent answers to our questions and be able to explain them. One of the most advanced attempts to build such a system is SHRDLU. It is discussed below.

SHRDLU, developed by Winograd [1972, 1983], simulates the operation of a robot arm that manipulates toy blocks on a table. The system accepts statements and commands as well as answers questions about the state of its world, and uses reasoning to decide its actions. The implemented system consists of four basic elements: a parser, a recognition grammar for English, programs for semantic analysis (to change a sentence into a sequence of commands to the robot or into a query of the database) and a problem solver (which knows how to accomplish tasks in the blocks world). The main idea of the implementation is that meanings of words, phrases, and sentences can be embodied in procedural structures and that language is a way of activating appropriate procedures within the hearer. SHRDLU's grammar consists of pieces of executable code. For example, a rule saying that a sentence is composed of a noun phrase and a verb phrase S —> NP VP, was embodied in the MICRO-PLANNER procedure.

```
( PDEFINE SENTENCE
        ((( PARSE NP ) NIL FAIL )
         (( PARSE VP ) FAIL FAIL RETURN )))
```

This program uses independent procedures for parsing a noun phrase and a subsequent verb phrase. These can call other procedures. The process FAILs if the required constituents are not found. Once rules produced a syntactic parse tree, separate semantic rules are applied to build the semantic representation, which is then used to manipulate the blocks world or to answer questions. Meaning of the words and sentences is a program which, when run, will produce the desired results. With such special procedural representations for syntactic, semantic, and reasoning knowledge, SHRDLU is able to achieve unprecedented performance.

It is possible to have an extensive model of the structures and processes allowed in the domain, because the system operates within a small domain. Knowledge about the state of the world is translated into MICRO-PLANNER assertions, and manipulative and reasoning knowledge is embodied in MICRO-PLANNER programs. For example, the input sentence *The pyramid is on the table* may be translated into a two arguments assertion (ON PYRAMID TABLE). SHRDLU's grammar is based on the notion of systemic grammar, a system of choice networks that specify the unordered features of syntactic units like clauses, groups and words, their functions, and their influences on other units. The parsing process lookes for syntactic units playing a major role in meaning, and the semantic programs are organized into groups of procedures that are applicable to a certain type of syntactic unit. In addition, the database definitions contain semantic markers (calls to procedures) that recognise semantically incorrect sentences. These semantic programs can also examine the context of discourse to clarify meanings, establish pronoun referents, and initiate other semantically guided parsing functions.

The significance of SHRDLU in NLP research lies in the demonstration of incorporating models of human linguistic and reasoning methods in the language understanding process. In opposition, Wilks [1973] has argued that SHRDLU's power comes from the use of problem-solving methods in a simple and closed domain, thus eliminating the need to address some of the more difficult language issues.

3. DISCUSSION OF THE ROLE OF SYNTAX AND SEMANTICS

The previous section reviewed some NLP systems with the purpose to give the reader an understanding of what kind of problems this area of artificial intelligence deals with, and how important it is to establish the proper relation between the syntax and semantics of the processed language. This relation, one of the most heated controversies in NLP, is discussed in the current section.

Fundamental for NLP is the problem of having an adequate grammar which is able to recognize and generate in an efficient way each of an infinite range of sentences, correct from the point of view of syntax and semantics. Syntax defines how to construct sentences, clauses and phrases from particular words. Semantics decides about their meaning. Syntactical correctness is independent of meaning. For example, the sentence *He tries to rain* is syntactically correct, but it is hard to put any meaning into it. Meaning of the sentence depends on the degree of its plausibility. It indicates its relationship to an external reality or an action to be performed on this reality. Meaning can be figured out even from nongrammatical sentences. For example, syntactically incorrect sentence *I wants go movie* can be understood. If number of errors, however, is too big, meaning is difficult or impossible to recognize. For example *I he like with after the street.* Sometimes the meaning is directly derived from the syntax, for example, by the construction *I wish I had* , we express a wish which we consider practically impossible to make come true.

Syntax is defined by a set of rules which describe the correct form of sentences, the sequence of the words, and the way in which they can appear. There is no general procedure to derive meaning from sentences. But one is able to do this in particular cases. For example, let us consider the sentence *Carl prepares good dinners.* The structure which matches this sentence is: **Proper Noun—Verb—Noun Phrase.** This structure describes the phenomenon: *somebody* performs some *action,* and there is an *object* which is the result of the action. One assigns: *somebody* to Carl, *action* to preparation, result of the *action* to dinner; the ending *s* indicating the progressive present tense tells that the action is being performed regularly. In this way one gets a *picture* of the phenomenon described by the sentence.

Another way of viewing syntax is thinking about it as a part of linguistic knowledge which is used in understanding new words and concepts. For example, when hearing the sentence, Carnap's example, *Pirots carulize elatically* one does not understand the meaning, but one agrees that it might be an English sentence. It is possible to recognize lexical category of individual words. For example, *pirot* is a candidate to be a noun, *carulize*– verb, and characteristic ending *ly* suggests that *elatically* may be an adverb or adjective.

Early NLP systems concentrated on syntax, because it is much easier to handle syntax than semantics. It can also be explained by big influence of Chomsky, who believes that exploring the syntax is a direct way to understand human mind.

In principle it is possible, after encoding into the parser every possible rule and creating a dictionary containing the words with lexical and other categories, to decide whether a sentence is correct or not from the syntactic point of view. Pure syntactic parsers are very inefficient and are not able to decide about semantic correctness of the sentence. They have no guidelines for choosing the best of the ambiguous parses. For example, they will assign the same structure for the sentence *Mary had a drink with lemon* and *Mary had a drink with John*. Syntactic parsers can be improved by taking advantage of semantics [Lesmo and Torasso, 1985], so that they inherently connect syntax with domain dependent semantics. This perhaps accounts for the fact that for each domain a new parser is written.

The task of recognizing semantically correct sentences is much more complex. It is not possible to give such a clear answer *yes* or *no* as in the case of syntactical correctness. Meaning depends on context, pragmatics and even on the fantasy of speakers. Awkward sentences can be bound through the power of flexible interpretation, metaphor and analogy. Chomsky's example *Colourless green ideas sleep furiously* could be explained, for example, in terms of fresh ideas which somebody wants to introduce, but so far does not succeed; *green* ideas are unripe ideas; some ideas are not that much interesting– *colourless*. A particular interpretation chosen from many possible ones reflects a person's character, his attitude towards others, or his emotional state.

Many recent NLP systems use the semantic parsing approach based on Conceptual Dependency theory developed by Schank. Given that individual words of our language denote concepts which have been created from observations of the environment (nouns and verbs) one can consider syntax as a means to express more complex relations between different concepts. So although semantics plays a much more important role than syntax, one should not forget about syntax, which can be very useful when deciding about semantic correctness of the sentence. Without syntax, a program would miss distinctions that have a major impact on meaning. For example,

John stopped to help Mary.
John stopped helping Mary.

Problems which are difficult or beyond the capabilities of purely syntactic parsers:

1. Word-sense ambiguity;
2. Structural ambiguity (prepositional phrase attachment);
3. Ill-formed input;
4. Metaphor;
5. Anaphora;
6. Pragmatics;

...

Problems which are beyond the capabilities of purely semantical parsers:

1. Loss of meaning when it depends on specific syntactical structure;
2. Difficulties of defining semantics.

There is a problem of giving a definition of semantics. There are not that many cases where semantics is precisely defined, as for example, in the Vienna Definition of Software Specification. Here semantics is understood as a function S:

$$S: L \rightarrow D$$

where L is a programming language as a set of programs, instructions, expressions, etc., D is a set of their *meanings* – denotations (i.e., functions operating on states of an abstract machine.) The denotation of an expression is a function that assigns values to the states, for example, Boolean values. The semantics function is defined by structural induction. Denotation of each complex syntactic object is described by composition of denotations of its components. So semantics is here the result of executing the instructions on a given machine. It would be difficult to define semantics in this way in the case of natural language because of the problem of defining the meaning of our utterances as a result of state changes. The difference between formal and natural languages is that formal languages are static and are the result of an explicit decision, which explains the precision of the semantics definition. Natural languages are dynamic and they change a great deal with time and growth of our knowledge about the world.

Linguists are concerned with the question of whether it is in principle possible in the case of natural languages with infinite scope to give a semantic description of the entire language, since it should be based on the knowledge humans have about the world and the society. This is said to be possible by some linguists if the problem is broken into parts [Vasiliu, 1981]. One can describe the semantics of natural language by producing a range of partial descriptions, each of which describes the semantics of a subset of the language. This position suggests that it is in principle impossible to design one consistent fully adequate grammar. Particular grammars describe only some features of the language. By narrowing the domain of a hypothetical system, one has chances of describing the language better. There is a consensus between this and the experience with NLP systems.

Without a theory of semantics when developing NLP systems, one will have to define and handle semantics locally taking into consideration the specific domain.

4. CONCEPTUAL DEPENDENCY THEORY

Some time ago it was suggested that meanings of words or sentences should be represented in a canonical manner, building them up of some small set of primitives, just as chemical substances are built of chemical elements. The best known and widely accepted attempt at a canonical representation is the Conceptual Dependency (CD) formalism developed by Schank as a reductionistic case frame representation for common action verbs. It tries to represent every action as a composition of one or more primitive actions, plus intermediate states and causal relations. Two identical actions expressed in different ways should have the same representation of the meaning. The number of semantic primitives (acts or states) should be small. Originally there were 11 primitive acts:

PTRANS	Transfer of the physical location of an object
PROPEL	Application of physical force to an object
ATRANS	Transfer of an abstract relationship
MTRANS	Transfer of mental information between people or within a person
MBUILD	Construction of new information from old
INGEST	Bringing any substance into the body
ATTEND	Focusing a sense organ
SPEAK	Producing sounds of any sorts
GRASP	Grasping an object
MOVE	Moving a body part
EXPEL	Pushing something out of the body

It is assumed that every sentence describes some event which CD tries to represent according to the schema ACTOR, ACTION performed by ACTOR, OBJECT that the ACTION is performed upon, and DIRECTION in which that ACTION is oriented. Relations between concepts are called dependencies. For example, canonical representations of the sentences *John gave Mary a book* and *Mary took a book from John* are:

```
[ ATRANS                        [ ATRANS
 relation:  possession           relation:  possession
 actor:     John                 actor:     Mary
 object:    book                 object:    book
 source:    John                 source:    John
 recipient: Mary ]               recipient: Mary ]
```

They describe the fact that a book was transferred from John to Mary; John had it before the action took place, Mary has it after the action and John no longer has it after the action.

Examples of primitive states are:

Mary HEALTH (-10.)	Mary is dead.
John MENTAL STATE (+10.)	John is ecstatic.
Vase PHYSICAL STATE (-10)	The vase is broken.

The number of primitive states is much larger than the number of primitive acts. States and acts can be combined. For example, the sentence *John told Mary that Bill was unhappy* is represented

John MTRANS (BILL BE MENTAL-STATE (5))to Mary.

An important class of sentences involves causal chains. There are 5 important rules that apply to CD theory:

1. Actions may result in state changes.

2. States can enable actions.
3. States can disable actions.
4. States (or acts) can initiate mental events.
5 Mental events can be reasons for actions.

These are fundamental pieces of knowledge about the world, and CD includes a diagrammatic pictorial shorthand representation of each (and combination of some) called causal links. Any implicit information in a sentence is made explicit in the representation of the meaning of that sentence. It is not clear when we should stop deepening such diagrams.

Schank's goal was to develop the system which would be able to perform such tasks as machine translation, paraphrasing, question answering and story understanding. By understanding stories, he meant understanding the relationship that one sentence has to another. Thus in order to handle texts (to make inferences and to connect sentences together) Schank and his colleagues introduced a few additional concepts:

script a sequence of standard situations, events; using scripts, it is possible to infer missing information in the text; it is a specific type of knowledge people possess about the world;

plan a sequence of actions which people are going to perform to achieve some goals; they are used when a person can not make sense of new input; it helps to solve new problems or deal with unexpected information; a knowledge of planning helps an understander to comprehend someone else's plan;

theme identifying a top level goal, that somebody is operating under allows one to predict the pattern of goals that he will pursue.

Schank's CD is widely used in many NLP systems, for example, in SAM developed by Cullingford, PAM developed by Wilensky - [Schank and Riesbeck, 1981], and also in learning systems like GENESIS developed by Mooney and DeJong [1985]. CD has some basic difficulties. It emphasises the meaning of an action verb in terms of its physical realisation. For example, *kiss* is reduced to MOVE lips to lips. Also the claim that CD preserves the whole meaning is clearly false. For example, in some situations the sentences *John gave Mary a book* and *Mary took a book from John* may have a slightly different meaning if Mary did not want the book (first sentence) or John did not want to give it to her. It is also not clear in which way CD solves the problem of different scope of meaning of the words in different languages (see sec.2.1 point

3.) Would it build the same representation for the sentences *Ich liebe diesen Wald* in German, and *J'aime ce foret* and *J'aime ce bois* in French? Nevertheless, CD is currently the best known formalism to represent the meaning of words and sentences, and many artificial intelligence researchers use its basic ideas.

5. MACHINE LEARNING SYSTEMS USING NATURAL LANGUAGE INPUT

A few systems combining NLP and machine learning were developed. Three of them, namely NANOKLAUS, Katz and Winston's system, and GENESIS, are discussed below.

5.1. NANOKLAUS

NANOKLAUS was developed by Haas and Hendrix [1983] as a pilot for KLAUS, a system aiding users in acquiring information. KLAUS was supposed to conduct conversations in English (limited to a very specific domain), retrieve and display information conveyed by the user, and learn new concepts (and their place in the hierarchy of already known concepts) and linguistic constructions as supplied by the user. The emphasis was on the problem of learning concepts and language simultaneously. NANOKLAUS has a fixed set of syntactic and semantic rules covering a small subset of English. Its grammar consists of a number of very specific rules for processing various types of sentences; it works by simple pattern matching. For example, the rule

<SENTENCE> → <PRESENT> THE <KNOWN-COUNT-NOUN> |
(DISPLAY <KNOWN-COUNT-NOUN>)

is used to match such inputs as:

What are the ships ?
Show me the officers.
List the carriers.

The metasymbol <PRESENT> matches the the italicised portion of these inputs, THE matches *the*, and <KNOWN-COUNT-NOUN> matches the last word in each example.

Some syntactic structures are used principally to introduce new concepts. Only the feature of learning new concepts is present, the system can not learn new linguistic structure. For example, the structure

<SENTENCE> → <A> <NEW WORD> <BE> <A> <KNOWN-COUNT-NOUN>
A carrier is a ship

means for NANOKLAUS that a new concept is being introduced. After recognizing the concept-defining pattern the system generates new entries in its lexicon, *carrier* in the example above, and creates a new predicate for it in the system's knowledge base

(ALL X) (CARRIER (X)→ SHIP (X))

in the example. Learning a *new individual* means creating a new constant term relating to one of the sorts, for example KITTYHAWK (JFK.) Learning a new verb, *command* for example, implies creating a new predicate with the proper number of argument positions; the system also constrains the domains of those arguments by such assertions as

(ALL X Y) (COMMAND (X Y)→ (OFFICER (X) AND SHIP (Y))).

NANOKLAUS allows queries in both active and passive voice. It translats clauses into internal structures of the form: (VERB-PREDICATE Arg1 Arg2 Arg3) using information about permissible syntactic patterns in which the clause's verb can occur. There are 13 such patterns. Modal verbs are not handled. Originally the syntactic category < KNOWN-COUNT-NOUN > contains only count nouns associated with seed concepts, such as *thing, person, physical object*, and other. The system asks questions to determine relationships between the sorts of objects that these new concepts are and other sorts of objects that are known to it. Response generation is accomplished by means of preprogrammed phrases and templates. It builds a hierarchical knowledge base by conversing with a user. It is an example of machine learning system employing learning by instruction strategy.

5.2. KATZ and WINSTON's SYSTEM

Katz and Winston [1982] developed a parser for parsing and generating English based on commutative transformations, which is currently used for natural language interaction with Winston's analogy learning program [Winston, 1981] and Binford's ACRONYM [Binford et al., 1982]. A semantic net is the common internal representation shared by the parser, the learning system, and the generator. The three step language generation procedure:

1. Converts a network fragment into kernel frames;
2. Chooses the set of transformations;
3. Executes the transformations, combines the altered kernels into a sentence, performs a pronominalization process (by comparing current and previous lists of noun phrases) and produces the correct English statement.

Parser translates from English into semantic net relations, and vice-versa. Each relation in the net is implemented as a frame (occupied by a noun or by an embedded relation), a slot in the frame (occupied by a verb or a preposition), and value in the slot (occupied by a noun, an adjective or an embedded relation). Nodes and relations in the net are created using the function RELATION. To describe the algorithm we use all the same example *Othello did not want to kill Desdemona because he loved her.* The relation, for example,

```
( RELATION OTHELLO
       WANT
       ( RELATION OTHELLO KILL DESDEMONA )
```

has representation

```
( WANT-1  ( FRAME ( OTHELLO ) ( SLOT ( WANT ) ( VALUE ( KILL-1 ) )
( KILL-1  ( FRAME ( OTHELLO ) ( SLOT ( KILL ) ( VALUE ( DESDEMONA ) )
```

Let us describe the generation procedure in more detailed way.

Step 1.

Each kernel element of the semantic network is turned into a corresponding kernel frame constructed from instantiated templates for noun and verb, according to the structure:

$$NT^{initial}\ NT^{agent}\ VT\ NT^{goal}\ NT^{theme}\ NT^{final}$$

where $NT^{initial}$ and NT^{final} are noun-templates that will be transformed later into the sentence's *initial* and *final prepositional phrases*; NT^{agent}, NT^{goal}, and NT^{theme} are noun-templates that play, respectively, the roles of *agent* (an entity that causes the action to occur), *goal* (the recipient or the beneficiary of the action), and *theme* (the entity that undergoes a change of state or position.) Examples of NT and VT:

NT = ((prep **(from out of)**) (det the) (adj nil) (noun **darkness**))
from out of the darkness
VT = ((aux1 **could**) (aux2 **have**) (aux3 nil) (verb **noticed**))
could have noticed

Two templates of the same type can be combined (concatenation or conjunction.) For example,

NT = ((prep nil) (det nil) (adj nil) (noun **Othello**) (conj **and**)
(prep nil) (det nil) (adj nil) (noun **Desdemona**))

Out of the instantiated templates two kernel frames are built: a matrix kernel frame (MKF) and an embedded kernel frame (EKF.) The EKF is used to construct sentences with embedded clauses- its position is indicated by the word *it* in MKF. For example

MKF = ((NT^{agent} ((prep nil) (det nil) (adj nil) (noun **Othello**)))
(VT ((aux1 nil) (aux2 nil) (aux3 nil) (verb **wanted**)))
(NT^{theme} ((prep nil) (det nil) (adj nil) (noun **it**)))

EKF = ((NT^{agent} ((prep nil) (det nil) (adj nil) (noun **Othello**)))
(VT ((aux1 nil) (aux2 nil) (aux3 nil) (verb **kill**)))
(NT^{theme} ((prep nil) (det nil) (adj nil) (noun **Desdemona**))))

Step 2.

Kernel frames are converted into matrix transformation frames (MTF) and embedded transformation frames (ETF) for the application of transformations. This involves conversion of the noun-templates into word strings, separation of each auxiliary verb from its affix, insertion of certain dummy slots used by certain transformations. The noun phrases of the transformation frame are derived from the noun-templates of the kernel frame. Each noun phrase has one of three fixed positions in the transformation frame: NP_1- position, $NP_{1.5}$- position, and NP_2- position. Noun phrases NP_1, $NP_{1.5}$, and NP_2 initially get their values from the templates NT^{agent}, NT^{goal}, and NT^{theme}, respectively. *Affix stripping* procedure separates each auxiliary verb from its associate affix. The affixes of the auxiliaries MODAL, HAVE, and BE are, respectively, *0*, *-en*, and *-ing*. For our example MTF has the following form:

MTF = ((COMP comp)(NP1 Othello)(TENSE past)(INFL infl)(AUX1 do)(NEG1 neg1)
(NEG2 neg2)(VERB want)(NP2 (it))

The network fragment determines which of 21 transformations (10 *connective* transformations prepare a MTF and ETF for combination; others, such as *negation*, *passivization* or *there insertion*, apply only to one frame) should be applied; they are part of a planning vocabulary, and they help to determine the meaning or focus the emphasis of a kernel sentence. The dictionary entry for any verb which may appear in a matrix clause contains a list of permissible transformations. An example of connective transformations is:

0–0–TO[1] *(John claims it) (John has written the letter) (John claims to have written the letter)*

Step 3.

All specified transformation are executed. Once they have been applied, purely syntactical adjustment operations are performed:

1. *Garbage–deletion* removes all unspecified elements;
2. *DO–deletion* deletes the auxiliary do when it immediately precedes a verb;
3. *Affix–hopping* recognizes situations in which verbs need affixes attached;
4. *N't–hopping* recognizes situations in which auxiliary verbs need n't attached.

In our example, all the adjustments except *N't– hopping* have an effect, producing the following result:

MTF= ((NP1 (OTHELLO))(AUX1 did)(NEG2 not)(VERB want)(NP2(it)))

ETF= ((INFL to)(VERB kill)(NP2 (DESDEMONA)))

Reading off the values in the adjusted transformation frames and substituting ETF at the joining point indicating by *it* in MTF we have final English form *Othello did not want to kill Desdemona because he loved her.*

Parsing is the reverse of generation (a given sentence is split into a set of kernel clauses; templates are filled out from left to right.)

Kats's parser is successfully used as front–end for Winston's analogy learning program which elaborates a set of rules from input stories using a version of Frame Representation Language (see Rule–1 below.) For example, from the story

MA is a story about Macbeth, Lady–macbeth, Duncan, and Macduff. Macbeth is an evil noble. Lady–macbeth is a greedy, ambitious woman. Duncan is a king. Macduff is a noble.
Lady–macbeth persuades Macbeth to want to be king because she is greedy. She is able to influence him because he is married to her and because he is weak. Mackbeth murders Duncan with a knife. Macbeth murders Duncan because Macbeth wants to be king and because Macbeth is evil. Lady–macbeth kills herself. Macduff is angry. Macduff kills Macbeth because Macbeth murdered Duncan and because Macduff is loyal to Duncan.

---

[1]Names of the transformations have following structure: COMP–NP1–INFL; COMP is inserted in the beginning of the structure and INFL– before the first auxiliary verb; NP1 receives its value from the frame NF^{agent}. COMP stands for complement, NP– noun phrase, and INFL for inflection.

it infers that the weakness of a noble and greed of his wife causes the noble to want to be king, and created the following rule:

```
 RULE–1
if
          [LADY–4 HQ GREEDY]
          [NOBLE–4 HQ WEAK]
          [[NOBLE–4 HQ MARRIED]] TO LADY–4]
then
          [NOBLE–4 WANT [NOBLE–4 AKO KING]]
case MA,
```

where HQ stands for *has quality*, and AKO– *a kind of.*

This rule can subsequently be used to make inferences about possible agent's goals in the story analogous to the processed one.

Katz's parser can process multi–clause embedded sentences. The problem with transformational grammars in general is that they are good for text generating rather than for parsing [Winograd, 1983]. They can produce from a given semantic network a range of different in sense of surface structure sentences having almost the same meaning. Transformations base mostly on syntactic features (one exception in Katz's parser is *Dative Movement* transformation). Because of predominant syntactic nature, the parser probably generates sentences as *She has a big nose therefore next week we are leaving for Chicago.* It is also unclear in which way compound sentences are processed.

5.3. GENESIS

Another example of a machine learning system using natural language is GENESIS [Mooney and DeJong, 1985] (GENeralizing Explanations of Stories Into Schemata.) An input story in natural language is processed by a parser, an adaptation of McDYPAR [Dyer, 1983], into a conceptual representation, a case–frame representation which uses some Conceptual Dependency primitives and predicates.

The understanding ability of the system concentrates on constructing a causal chain of actions by inferring missing information and causally connecting inputs together. This is done on the conceptual level by comparing a model built from the input story and suggested schemata.

The difficult problem of choosing the subset of suggested schemata from all of schemata is solved by attaching a schemata class to the input. It avoids the combinatorial explosion of the search, but also means lack of a good mechanism for recognizing the class of the schema for the input story. GENESIS is able to produce new schemata within one schema class when it finds a novel way, new set of actions, which led to achieving one of the thematic goals. In this lies its learning ability. It is incapable of understanding stories which do not suggest known schemata, hence it rearranges rather than produces knowledge. In other words it learns in incremental fashion depending heavily on background knowledge it possesses.

Currently the system has 91 schemata of one of two schema types: kidnapping of an individual and holding them for ransom, and burning somebody's own building to collect the insurance. It does not *understand* the concepts it knows; it is not able, for example, to answer the question *What is money ?*, but it can answer that money is something valuable for people. GENESIS has a very restricted natural language– it recognizes 100 words concerning its two known schemata. It was difficult even for the author to remember what are the system's constraints on the language.

6. COMPARATIVE SUMMARY OF DISCUSSED SYSTEMS

The systems discussed in this paper are representative of various directions of research on NLP. This section summarises their properties, compares them pointing out their merits, demerits and most characteristic features. The comparison is done at an abstraction level that permits us to ignor the fact that they address slightly different problems and serve different purposes.

Wilks's system performs sentence–by–sentence translation from English into French, and vice–versa. Syntactical and semantical knowledge is mixed in templates. The system uses a static set of handcrafted language specific rules.

MOPTRANS is a multilingual parser translating short newspapers articles. It builds an intermediate conceptual representation of the text which results in meaning–preserving translation. The rules it uses operate on words' classes. Syntactical and semantical knowledge is represented by two separate, intercommunicating modules.

LUNAR retrieves information about moon rocks from static data base. Its vocabulary of 3500[2] enables practical usage of natural language.

SHRDLU uses linguistic and reasoning methods in its language understanding to deal with its toy world (boxes, pyramids, blocks, cubes.) It reasones about its actions and explaines its previous behaviour. Procedural semantics resultes in its high performance. It coveres a large subset of English.

NANOKLAUS only approaches an interesting problem of learning new concepts and syntactic constructions from the user utterances. Its restricted dialogue is limited to ships. Its grammar uses simple pattern matching. The system can be viewed as a user friendly interface for building a knowledge base.

Katz's parser, serving as a front–end for Winston's analogy learning program, is syntax oriented and therefore fairly general. It applies previously learned rules to analogous situations. It handles fairly complex sentences.

GENESIS demonstrates explanation–based learning on a small 100 word vocabulary to process short stories about kidnapping and dishonestly collected insurance. It learnes how to achieve goals in a novel, more efficient way. Based on known schemata it inferres missing information and causally connectes input sentences together using a large amount of handcrafted domain knowledge.

The analysed systems have different practical value: LUNAR serves well in real–world situations, Katz's and Winston's system and GENESIS[3] are useful for demonstrative and research purposes, and the rest is something in–between; closer, however, to the second type. LUNAR demonstrates high performance which can be explained by the static and thematically limited data base it workes with. This is an exceptional situation, especially if learning is involved.

If one desires a practical system involving both communication in natural language and learning, for example, a Winograd–style robot performing some actions in quasi–natural environment, then features reflecting different aspects of humans' learning and performing should be creatively combined within one system. In particular, we mean features present in the discussed systems: natural language dialogue,

[2]Different lexical forms of the same word like, for example, *do does did done*, are not counted.

[3]These two systems concentrate on learning aspects; natural language input facilitates only the communication with the systems.

procedural knowledge and reasoning, high performance (SHRDLU), fairly unconstrained language, higher level analogy learning (Katz's and Winston's system), inferring missing information, learning more effective ways to achieve goals, expectation-driven inferences about typical situations in order to understand new ones (GENESIS).

It is a general trend that learning systems developed recently extract information from natural language input by parsing it into some conceptual representation. Inferences are then made on the conceptual level. The results are later transformed into natural language sentences. Many systems with natural language input use the Conceptual Dependency (CD) formalism.

CD formalism has proven itself in many NLP and learning systems. It has, however, the disadvantage of loosing some information contained in natural language utterances. Its representation is less meaningful than the entire input, which is exactly opposite to the way it is in the human mind. The claim that CD provides for unified general representation is not true. CD primitives are not sufficient for new domains; one has to develop new, adequate ones.

During the past decade several parsers were developed, for example, Katz's parser at MIT [Katz, 1980], [Katz and Winston, 1982], ELI at Yale [Schank and Riesbeck, 1981], or McDypar at Yale [Dyer, 1983]. Despite the claims about their generality they are task and/or domain dependent. Processing natural language progressed from the time of early NLP systems, but the fact that parsers operate well only in restricted domains and on very small (in the sense of English coverage and/or vocabulary) subsets of natural language remains unchanged.

NLP and machine learning use a large variety of knowledge representations which are often task dependent. People seem to store knowledge in a unified and still very efficient way. Knowing how to do this would facilitate development of an efficient, and at the same time general or easily adaptable to specific needs parser.

Both NLP and machine learning suffer a lack of good concept formation and representation theory. This fact is revealed, for example, in poor results of machine translation. Second generation machine translation systems, mapping text from one language to another without building the intermediate language-free meaning representation, are in principle incapable of improving the quality of translation. This is because they perform mapping on a lexical, instead of a conceptual, level. The approach undertaken in MOPTRANS seems to be more fruitful.

We do not have good understanding of the correspondence between concepts stored in our memory and words in our language. Similiar words in different languages do not convey the same meaning. In the examples in sec. 2.1.2 the concept *EXPLODE-BOMB* was translated into English as *bomb* (*...injured by bomb*), which is correct, and into German as *Bombe* (*...mit einer Bombe verwundet*), which should be correctly translated as *Bombenexplosion.* (See also sec. 2.1 point 3- words with different scope of meaning.)

The relation between a word and a concept denoted by it is probably culture dependent. Certainly, many words denote the same concepts in different languages, for example *mother* or *milk.* Likewise, sometimes it is difficult to remember in what language we were given some information (it may also be evidence for the fact that there exists higher level *mentalese*). Some concepts, however, are specific for one culture. When talking with friends of mine, we even do not notice switching between Polish, English and Russian. This may indicate that there are things which one can express in the best way in a specific language.

Hierarchy of shared concepts depends on the culture. In order to make a good translation from one language to another one needs to have broad knowledge of both cultures, and to have a *measure* of similiarity between particular concepts in the two languages.

Language constantly undergoes many changes. New words and concepts are created, words change their meanings. Sometimes it is difficult to explain some expressions because the situation which provided for their development is forgotten (as in the example in sec. 2.1 point 2.- not every Pole knows the history of it.) The question, as to whether we are able to reconstruct these processes and find the rules which affect the current state of the language, is to be answered.

7. CONCLUSION AND SUGGESTED RESEARCH

The primary purpose of this paper was to investigate existing systems with natural language input, and to evaluate their usefulness for machine learning research. An underlying secondary purpose was to explore the possibility of communication in natural language with machine learning programs developed in our AI Laboratory at the University of Illinois. As indicated above, currently there is no existing system which can serve as front-end for our programs. In this context we would suggest short-term and long-term research topics.

As to the short-term research, an interesting topic would be to integrate natural input with our currently most powerful, multi-purpose program INDUCE-4 [Mehler, Bentrup and Riesedel, 1986] [Hoff, Michalski and Stepp, 1983]. This program learns incrementally structural descriptions of object classes from examples. It is capable of constructing new attributes not present in the original data. This program is of particular interest to us because there are many application domains where objects to learn about have an intrinsic structure, and cannot be adequately characterised by attributes only.

In order to provide an integration one might build a parser translating English sentences into the Annotated Predicate Calculus (APC) used in the program. Examples of input sentences in natural language and desirable output in the form of APC expressions:

Input: *Trains going to Chicago in the afternoon have 4 to 6 green cars.*
Output: [∀ x: TRAIN(x) & DESTINATION(x)=Chicago & TIME_OF_DEPARTURE(x)=afternoon]
=> [NUMBER_OF_CARS(x)=4..6] & COLOR_OF_CARS(x)=green]

Input: *Dobermans and schnautzers are good watching dogs.*
Output: [∀ x: BREED(x)=doberman v schnautzer]
=> [TYPE_OF_DOG(x)=watching & QUALITY_OF_DOG(x)=good]

In order to generate such predicates as COLOR or NUMBER_OF_CARS, not present in the input sentence, the system should be equipped with the knowledge of hierarchy of concepts (words) along with relations among them. Unknown concepts (words) and their relation to the known ones could be learned interactively from users. Such an intelligent parser will release users from tedious translation of the input examples into formal language or relational tables.[4]

The system could be built on top of McDypar- an expectation-base, left-to-right, bottom-up conceptual parser. Following changes and extentions of McDypar are suggested: module enabling learning new concepts (words), module handling discourse,[5] procedures handling complex sentences (conjunction, subordinate clauses, etc.), routine handling passive voice transformation, feature of recognizing noun groups with nouns as modifiers, resolving pronouns references, special handling *wh-*[6] questions. ; all the modules will be controlled by routine performing the main task of translating pieces of natural language input into APC expressions.

Considering long-term research, the development of an adequate concept formation and representation theory is crucial for further progress in both fields NLP and machine learning. The concept representation approach treating concept as a static collection of attributes is not sufficient to explain the richness of natural language conceptual structure [Murphy and Medin, 1985]. The dynamic nature of concepts may be captured by representing them using two components : a static *base* and an *inferential concept interpretation* [Michalski, 1986]. The *base* contains the easily-definable, typical meanings, and *inferential concept interpretation* matches representation with observations by applying various types of inference using context and background knowledge. The idea of such two-tiered concept representation needs to be explored in more detail. Concept representation in machines should allow making inferences in an easy, effective and natural way.

An adequate concept formation and representation theory would enable attacking another important research task, namely, finding relation between words of our language and concepts denoted by them. Currently known semantic representations, such as CD, are not sufficient because they ignore the fact that concepts in different languages often refer to different things. An evidence for this is, as we have shown in the sec. 2.1.2 and 6., sometimes awkward surface structure of sentences translated by MOPTRANS.

[4]Some examples, however, would be easier to represent in the form of the relational tables- users will make their choice which representation form of the input they prefer.

[5]This feature is of a big importance for our recently undertaken robotics Intelligent Explorer (IEX) project.

[6]*Wh-* questions start with such words as *which, what, why*.

Difficulties in NLP and machine learning research reflect complexity of the task of development of systems capable of learning from natural language input. This is, however, the right direction if one considers computers as intelligent humans' assistants.

REFERENCES

(1) Barr, A. and Feigenbaum, E.A., "The Handbook of AI", vol.I, III, HeurisTech Press, Stanford, CA, William Kaufman, Inc., Los Altos, CA, 1981

(2) Bulas, K., Lawrance, L.T. and Whitfield, T., "The Kosciussko Foundation Dictionary Polish-English", N.Y., 1967

(3) Charniak, E., "Introduction to Artificial Intelligence", Addison-Wesley Publishing Company, 1984

(4) Chomsky, N., "Aspects of the Theory of Syntax", The MIT Press, 1965

(5) Dyer, M., "In Depth Understanding", MIT Press, Cambridge, MA, 1983

(6) Frederking, R.E., "Syntax and Semantics in Natural Language Parsers", CMU, Dpt. of CS, CMU-CS-85-133, 1985

(7) Haas, N. and Hendrix, G.G., "Learning by being told: Acquiring Knowledge for Information Management" in "Machine Learning-An Artificial Intelligence Approach", vol.I, Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Tioga Publishers, 1983

(8) Hayes, P. and Carbonell, J., "A Tutorial on Techniques and Applications for Natural Language Processing", CMU, Dpt. of CS, CMU-CS-83-158, 1983

(9) Hoff, W., Michalski, R. and Stepp, R., "INDUCE-2: A Program for Learning Structural Descriptions from Examples", UIUC, Dpt. of CS, UIUCDCS-F-83-904, ISG 83-4, 1983

(10) Katz, B., "A Three-Step Procedure For Language Generation", MIT, Dpt. of CS, AI Memo No.599, 1980

(11) Katz, B. and Winston, P.H., "Parsing and Generating English Using Commutative Transformations", MIT, Dpt. of CS, AI Memo No. 677, 1982

(12) Lesmo, L. and Torasso, P., "Weighted interaction of syntax and semantics in natural language analysis" Proceedings of the 9-th International Joint Conference on Artificial Intelligence, Los Angeles, 1985

(13) Lytinen, S.L., "The Organization of Knowledge In a Multi-lingual Integrated Parser", Yale, Dpt. of CS, Yaleu/CSD/RR#340, 1984

(14) Mehler, G., Bentrup, J. and Riedesel, J. "INDUCE.4: A Program for Incrementally Learning Structural Descriptions from Examples" Department of Computer Science, University of Illinois, Urbana, 1986 (in preparation)

(15) Michalski, R., "A Theory and Methodology of Inductive Learning" in "Machine Learning- An Artificial Intelligence Approach", vol.I, Michalski, R., Carbonell, J.G. and Mitchell, T.M.(editors), Tioga Publishers, 1983

(16) Michalski, R., "Two-tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness", an invited paper for the Allerton Conference on Analogy and Similarity, June, 1986

(17) Mooney, R. and DeJong, G., "Learning Schemata for Natural Language Processing", Proceedings of the 9-th International Joint Conference on Artificial Intelligence, Los Angeles, 1985

(18) Murphy, G.L. and Medin, D.L., "The Role of Theories in Conceptual Coherence", Psychological Review, vol.92, Nr.3, 1985

(19) Schank, R., "The Role of Memory in Language Processing" in "The Structure of Human Memory", W.H. Freeman and Company, San Francisco, 1975

(20) Schank, R., "Language and Memory", Cognitive Science 4, p.243-284, 1980

(21) Schank, R. and Riesbeck, C.K, "Inside Computer Understanding", Lawrance Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1981

(22) Schank, R. and Burstein, M., "Modelling Memory for Language Understanding", Yale, Dpt. of CS, Research Report #220, 1982

(23) Sowa, J.F., "Conceptual Structures: Information and Processing in Mind and in Machine", Addison-Wesley Publishing Company, 1984

(24) Vasiliu, E., "Elementy semantycsnej teorii jesykow naturalnych" (in Polish.) Zaklad Narodowy im. Ossolinskich, 1981

(25) Wilks, Y.A., "An AI Approach to Machine Translation" in Schank and Colby (Eds.) "Computer Models of Thought and Language", San Francisco, Freeman, 1973

(26) Winograd, T., "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language", Ph.D. Thesis, MIT, Cambridge, 1971

(27) Winograd, T., "Language as a Cognitive Process, Vol.I: Syntax", Addison-Wesley Publishing Company, 1983

(28) Winston, P.H., (Ed.) "The Psychology of Computer Vision", 1975

(29) Winston, P.H., Binford, T.O., Katz, B. and Lowry, M., "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents", MIT, Dpt. of CS, AI Memo 679, 1982, revised 1983

(30) Winston, P.H., "Learning by Augmenting Rules and Accumulating Censors", ML Workshop, UIUC, 1983

(31) Woods, W., "Progress in Natural Language Understanding: an Application to Lunar Geology", AFIPS Conf. Proc. 42, pp.441-450, National Computer Conference, Montvale, N.J., 1973

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-F-86-953	2. ISG 86-10	3. Recipient's Accession No.
4. Title and Subtitle Critical Issues in Natural Language Processing and Their Importance to Machine Learning			5. Report Date August 1986
			6.
7. Author(s) Lucja Iwanska			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. N00014-82-K-0186 NSF DCR 84-06801 N00014-K-85-0878
12. Sponsoring Organization Name and Address Office of Naval Research, Arlington, VA National Science Foundation, Washington, DC Defense Advanced Research Project Agency, Arlington, VA			13. Type of Report & Period Covered
			14.

15. Supplementary Notes

16. Abstracts

Several representative natural language processing systems are reviewed and compared. The review is done from the viewpoint of issues related to the development of systems capable of learning from natural language input. Specifically, such issues are considered as representation of concepts, learning capabilities of the systems, the role of syntax and semantics, restrictions on language and domain, and tradeoff between generality and efficiency. It is shown that further progress in both fields, natural language processing and machine learning, depends on elaborating the theory of concept formation and representation.

17. Key Words and Document Analysis. 17a. Descriptors

Concepts Formation and Representation
Machine Translation
Natural Language Processing
Semantics
Syntax

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 37
	20. Security Class (This Page) UNCLASSIFIED	22. Price