A. R. D. Prasad
DRTC, Indian Statistical Institute

PROMETHEUS: An Automatic Indexing System

Abstract: An automatic indexing system using the tools and techniques of Artificial Intelligence is described. The paper presents the various components of the system like the parser, grammar formalism, lexicon, and the frame based knowledge representation for semantic representation. The semantic representation is based on the Ranganatban School of thought, especially that of Deep Structure of Subject Indexing Languages enunciated by Bhattacharyya. It is attempted to demonstrate the various steps in indexing by providing an illustration.

1. Introduction
   The present work is an attempt to build an automatic indexing system using the tools and techniques of Artificial Intelligence. Information is presented to the indexer in natural language form and it warrants that any attempt in building an automatic indexing system should incorporate Natural Language Processing (NLP) tools.

   Though, natural language syntax presents many problems, the real crux of NLP is semantics. Natural language is knowledge dependent. An NLP system should contain the knowledge of vocabulary, knowledge of syntax and also the knowledge of the specific domain.

2. Objectives
   - To design and develop an automatic subject indexing system.
   - To demonstrate that NLP approach has more potential in subject indexing as it attempts at semantics, unlike statistical techniques.

3. Hypotheses
   The general theory of Subject Indexing Languages (SIL) forms the basis for this work:
   1. The deep structure of SIL could be used as a semantic structure for expressive titles. In other words, the deep structure of subject indexing languages could serve as a Meaning Representation Language (MRL) for a natural language processing system.
   2. Conceptual dependency grammars and many other grammar formalisms are heavily verb oriented. However, in the case of NLP systems for subject indexing, they should be noun phrase oriented.
   3. Noun phrases that occur in a particular content description (expressive title) have a fixed role to play and have fixed relation to each other, irrespective of the noun phrase position in a particular sentence.
   4. A frame based knowledge representation model is well suited for arranging the roles of Noun phrases and their relation to other Noun phrases.

4. Overview of Prometheus
   Prometheus includes a bottom-up parser. The lexicon and syntax are built using

unification based grammars where linguistic features are expressed in PATR notation. The meaning representation language is based on the Ranganathan (1967) school of thought especially the Deep Structure of Subject Indexing Languages (Bhattacharyya, 1980), which he introduced in his Postulate based Permutated Subject Indexing (POPSI).

Prometheus takes expressive titles of documents as input. The natural language parser checks the syntax of input statements and generates noun phrases. The basic approach in subject indexing emphasizes the importance of noun phrases (Vickery and Vickery, 1992) unlike the Conceptual Dependency grammars (Schank, 1984) which is verb phrase oriented. The noun phrases that are generated by the parser are passed on to the meaning representation system, which is basically a knowledge representation model based on the deep structure of subject indexing languages.

4. 1 The Parser

The purpose of a parser is to compute syntactic structures out of the given natural language statements. A grammar formalism as a declarative description of language does not specify how syntactic structure is to be computed. The critical decision is to choose an appropriate one out of many parsing algorithms. There are three basic issues involved in parsing —— whether the parser should be bottom-up or top-down; whether it should be depth-first or breadth-first and the issue of storing the intermediate results. Prometheus has used left-corner, bottom-up parser (Gazdar and Mellish, 1989).

4. 2 Syntax

One of the significant changes in the programming field is the shift from procedural definition of semantics of programming languages to declarative description of denotational semantics. The change holds true even with natural language processing.

The declarative paradigm that is adopted is Context-Free Phrase Structure Grammar (CF-PSG) and to be specific the Definite Clause Grammars and PATR. PATR in particular has become potential lingua franca for NLP work and many other grammar formalism can be expressed in it.

Modern theories of grammar use features. The extent and sophistication of feature based granulars have grown massively in 1980’s. In feature theoretic syntax, atomic categories such as NP and V are replaced by sets of feature specifications. Feature specifications such as case (nominative, accusative), number (singular, plural), person (first second, third), etc., can be added to the grammar more elegantly.

In case of expressive titles of documents, it should be noted that they may not be complete English sentences. In other words, a set of syntactic rules are developed in order to accept titles of documents. One such rule expressed in terms of linguistic features in PATR notation is:

Rule NP --- \( \rightarrow \) [NP1, CONJ, NP2]:-

\[
\begin{align*}
NP &: \text{cat} \equiv np, \\
NP1 &: \text{cat} \equiv np, \\
NP2 &: \text{cat} \equiv np, \\
CONJ &: \text{cat} \equiv conj, \\
NP &: \text{np1} \equiv \text{NP1}:np, \\
NP &: \text{np2} \equiv \text{NP2}:np.
\end{align*}
\]
4. 3 Lexicon

Feature structures are now widely used to represent morphologic, syntactic and semantic information. Feature structures along with lexicon have permitted computational linguists to adopt very simple and compact rule system at the cost of passing almost all syntactic facts about language into the lexicon.

Word determination:
\[ W:\text{cat} = = = n, \]
\[ W:n = = = \text{determination}. \]

Word diagnosis:
\[ W:\text{cat} = = = n, \]
\[ W:n = = = \text{diagnosis}. \]

4. 4 Semantic Representation

In order to represent the meaning of given expressive titles, Prometheus uses a frame-based knowledge representation model based on the deep structure of indexing languages.

Each noun phrase that is passed on to the semantic system is analyzed in order to find out to which category it belongs. In other words, it is essential to recognize whether an isolate belongs to either entity, or property or action or speciator etc. In addition, the system performs standardization, modulation, assignment of indicators, generation of subject index entries following the syntax of the deep structure of subject indexing languages.

5. Formulation of Subject Index Entries

Prometheus adopts the following steps in order to arrive at the deep structure of subject indexing languages:

Step 1: Identification of syntactic categories of terms in an expressive title. Exclusion of articles, conjunctions, etc.

Step 2: Generation of syntactic structure.

Step 3: Construction of noun phrases which include uniterns, compound and composite terms.

Step 4: Standardization, identification of elementary categories for each noun phrase and assignment of indicators for each elementary category.

Step 5: Modulation

Step 6: Generation of subject entries using the syntax of deep structure subject indexing languages.

Illustrated Example:

The best way to understand how Prometheus generates deep structure of a given expressive title is to demonstrate it with the following title:


Step 1: Identification of syntactic categories.

The parser of Prometheus is a bottom-up parser i.e. it is data driven. The parser reads every word in the expressive title and looks for each word in the lexicon. In the lexicon, each
word is mentioned in PATR notation, in the following way:

Word determination :
  W : cat == = n,
  W : sem == = determination.

Word of :
  W : cat == = prep.

Word depth :
  W : cat == = adj,
  W : sem == = depth.

Word dose :
  W : cat == = n,
  W : sem == = dose.

In the lexicon, each word has an entry, describing the syntactic features and also, the word it should pass for the final syntactic structure. This is mentioned under the feature sem.

Step 2: Generation of syntactic structure

The parser of Prometheus attempts to apply the grammar rules specified in PATR notation. Thus, the adjectival phrases depth dose, satisfies the following syntactical rules

Rule NP ---> [ADJ, NPI] :
  NP : cat == = np,
  ADJ : cat == = adj,
  NPI : cat == = np,
  NP : np : adj == = ADJ : adj,
  NP : np : n == = NPI : np.

Rule NP ---> [NP] :
  NP : cat == = np,
  N : cat == = n,
  NP : np == = N : n.

These two rules state that a noun-phrase can contain an adjective followed by a noun. The rule for adjectival phrases is recursive, so that it can accept any number of adjectives to be proceeded before a noun as is the case in X-ray rotation therapy. Following is the syntactic structure produced by Prometheus for the given expressive title.


In this syntactic structure, it can be seen that Prometheus has accepted the given expressive title as a grammatical construct and indicated the syntactic categories of constituent words.

Step 3: Construction of Noun Phrases

This step is fairly straight forward once Prometheus generates the syntactic structure of the title. The system identifies each noun phrase and adds it to internal database for further semantic processing. It should be noted that the system generates compound words with a ' ' as connector. Thus the output of step 3 is:
Step 4: Identification of Elementary Categories

The noun phrases, thus generated by syntactic processing are then passed on to the frame based knowledge representation system based on the deep structure of subject indexing languages. Of the noun phrases generated, the system first picks up the word determination, and then searches for this word among the facts of the knowledge base until it encounters the fact,

\[ \text{value(determination, use, measurement).} \]

This fact indicates that the word determination is a non-standard term and should replaced by measurement. The next step is to identify the category of the word measurement, for which the system encounters, the following fact:

\[ \text{value(measurement, category, action-on-action).} \]

which says that the word measurement belongs to the category action-on-action. Then the next step is to identify the indicator for action-on-action, for which the fact

\[ \text{value(action-on-action, indicator, 8.2.9.9).} \]

informs the system that the indicator is 8.2.9.9. Thus the input word determination becomes 8.2.9.9 measurement. This fact will be added to a B-tree so that the systems produces subject heading, and places measurement depending on the ordinal value of 8.2.9.9.

The same steps are followed in producing the relevant substring for the phrase depth-dose, except that a connector [of] is added before the depth-dose, as the knowledge base contains the following fact:

\[ \text{value(depth-dose, connector, [of]).} \]

The phrase roentgen \textit{rotation} \textit{therapy}, is factored into roentgen and rotation \textit{therapy}. The word roentgen is replaced by X-ray using the fact

\[ \text{value(roentgen, use, X-ray).} \]

and

\[ \text{rotation \textit{therapy}, is replaced by rotation \textit{technique}, using the fact} \]

\[ \text{value(rotation \textit{therapy}, use, rotation \textit{technique}).} \]

Thus the system generates,

\[ \text{8.2.9.5 [using] X-ray and} \]
\[ \text{8.2.9.5 [using] rotation \textit{technique}} \]

To summarize, the following strings are added to the B-tree index

\[ [8.2.9.9, \text{measurement}] \]
\[ [8.2.9.9.5, \text{[of]}, \text{depth} \textit{dose}] \]
\[ [8.2.9.5, \text{[using]}, \text{X-ray}] \]
\[ [8.2.9.5, \text{[using]}, \text{rotation \textit{technique}} \]
\[ [8.2.9.9, \text{treatment}] \]
\[ [8.2.9.9.6, \text{[using]}, \text{ionization \textit{pocket \textit{chamber}}}] \]

Step 5: Modulation

The system attempts to find the broader terms, if any, for each and every phrase. The broader term, narrower term relation is expressed using kind-of or part-of relation. Thus, the term X-ray picks up a broader term radiation using the fact

\[ \text{value(X-ray, kind-of, radiation).} \]

from the knowledge base. Similar logic applies to the term
ionization" pocket" chamber
which encounters the following fact in the knowledge base
value(ionization"pocket" chamber, kind-of, ionization"chamber).
Each and every broader term along with its indicators and connectors are then added to the
B-tree.

Step 6: Generation of Subject Entries
The purpose of storing each term along with its indicators is to get the sorting order. The
final step involves simply printing the strings that are stored in the B-tree one after another.
Thus the expressive title determination of depth dose in roentgen rotation therapy using
ionization pocket chamber becomes

1 medicine 8 human"body 8.2 disease 8.2.9 treatment 8.2.9.5 [using] radiation 8.2.9.5
[using] rotation"technique 8.2.9.9 measurement 8.2.9.9.5 [of] depth"dose 8.2.9.9.6 [using]
ionization"chamber 8.2.9.9.6 [using] ionization"pocket" chamber.

6. Conclusion
Prometheus is not a fully automatic indexing system. It is an attempt towards building
such systems. Nevertheless, it substitutes human intelligence in all the required steps in subject
indexing except one -- preparation of the expressive title after carefully examining the
document title, abstract and text. Fully automatic indexing may take more time until the
natural language processing research offers definite results. However, Prometheus attempts
to generate subject strings from expressive titles presented in natural language form.
One of the significant aspects of Prometheus is its semantic representations. It attempts
to demonstrate that the deep structure of subject indexing languages could be used as a semantic
structure in subject indexing.

Prometheus is not without its limitations. The parser of Prometheus performs shallow
syntactic analysis. Though it uses feature based grammar, it does not include semantic features
of terms. It does only contain the syntactic features of terms. Most of the semantic task is
passed on to the frame based representation of knowledge. The question is, whether it is
advantageous to perform semantic processing along with syntactic processing or semantic
processing should follow syntactic processing, is highly controversial. However, an attempt
can be made in future to compare the results of both the approaches.

References
Dharwad: Karnatak University.
to computational linguistics. New York: Addison-Wesley.
Prasad, A.R.D. (1994). Application of computer based natural language processing tools and
University.
House.
Massachusetts: Addison-Wesley.