Serdica
Bulgariacae mathematicae
publicationes

Сердика
Българско математическо
списание

AUTOMATIC DATA EXTRACTION FROM SPECIALIZED TEXTS

PETER BARNEV, STEFAN KERPEDJIEV

Many difficulties connected with data input are caused by the fact that often data is scattered in certain texts instead of being separated. These texts are not quite arbitrary and consist of relatively limited number of linguistic structures. Typical classes of specialized texts are used for different purposes. Automatic data extraction from texts of a given class is proposed. This calls for a formal description of a significant part of the class and creation of algorithms for text analysis in conformity with that description. The difficulties due to different ways of data representation are surmounted by adequate data translation.

1. Introduction. The diversity of data sources and data representations gives rise to a great number of problems concerning the input of data. A large amount of human efforts is directed to data extraction and transformation and many mistakes are made while doing that. An approach to automating those processes in case of specialized texts used as data sources is proposed in this paper.

In many spheres of human activity texts are created by sticking to a certain scheme. For example, announcements for conferences, congresses, etc., autobiographies, some kinds of entries in encyclopaedias (e. g. of rivers, persons, etc.), weather forecasts, recipes, etc. could be considered as such texts. Texts of that type will be called specialized ones. Specialized texts are characterized by: they contain data which could be used in specific problems of a certain object field; text formation does not obey any strictly defined rules but is rather in accordance with some naturally formed requirements; the relatively strong specialization of the texts allows to describe them formally with the exception of some insignificant part of them.

We assume that data is represented in computers through an internal language which defines: data meanings; relation between the data values; and formats in which different data types are represented in the memory.

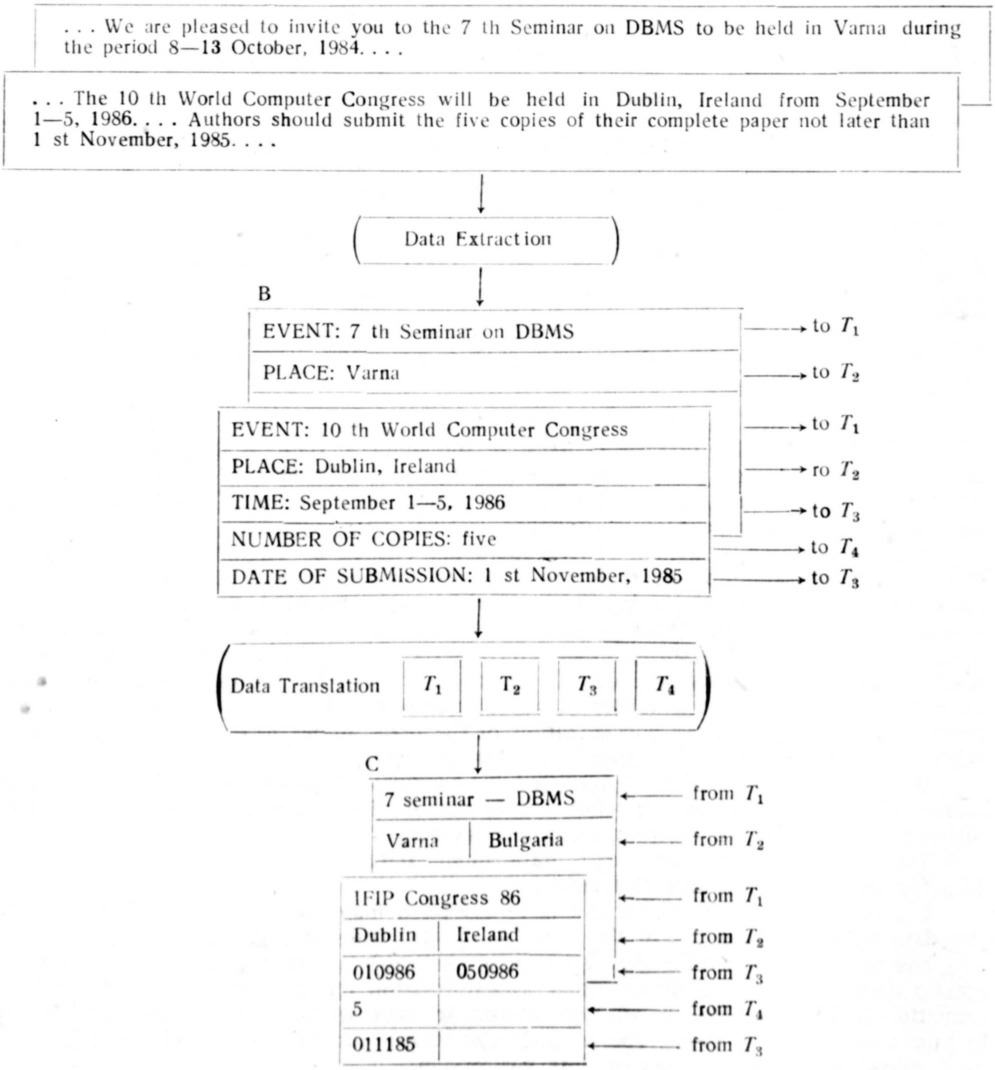
Two main problems arise with data input: data extraction from the text; data transformation into an internal format.

The essence of data extraction is to set up a mapping of the data meanings to the data values represented in the text ($A \to B$ transition in Fig. 1).

The values of a certain data type (e. g. — dates) are represented by various linguistic means which complicates their transformation into the internal format ($B \to C$ transition in Fig. 1). That is why an automatic data translation was proposed in [1]. In this case for each data type a particular translator (T_1, T_2, T_3, T_4 in Fig. 1) is used, allowing different means of data expression.

Data extraction and transformation into a unified linguistic form are usually performed by men and computers are used only in the last stage of their internal format transformation. The problem how computers could be involved into the whole process of data transfer from the text to the internal representation is considered here.

2. Data model. Each data item (d) belongs to a certain data type (t). Usually a data type is defined as a set of values and a set of operations on them. On the other hand, each data item presents a piece of real information and it is used for a definite



. . . We are pleased to invite you to the 7 th Seminar on DBMS to be held in Varna during the period 8—13 October, 1984. . . .

. . . The 10 th World Computer Congress will be held in Dublin, Ireland from September 1—5, 1986. . . . Authors should submit the five copies of their complete paper not later than 1 st November, 1985. . . .

A — Specialized Text
B — Text Interpretation
C — Internal Representation

Fig. 1. Data extraction and translation

purpose. That is why we say that it has a meaning (m). Each data item has a value (v), which can be expressed by different ways, e. g. the number of the fingers of the human hand could be expressed by "5", "five", etc. Further on each data value is considered to be unchangeable, but nevertheless it might be represented in various forms at the different stages of data fransformation. Thus, the concepts of data type and data value are meaningful here. These terms are considered more formally with algorithmic languages where by data value is meant one of its concrete representations and by data type is understood the set of representations obtained by applying certain rules. In order to distinguish the different meaning of the terms data value and data type two additional terms are introduced in our model. When treating formally the data type and data value we shall use the terms data format (f) and data representation (r), respectively. Thus, each data item could be considered as a triple of meaning, type and value: $d=(m, t, v)$ and one of its concrete representations could be obtained by specifying the format: $r=r\ (d, f)$.

Various formats are used for representing the data values and different transformations of them from one format to another are feasible in the limits of a certain data type. The set of such formats F may be considered as a union of two subsets F' and F'' ($F=F' \cup F''$). F' contains formats which have a natural origin (e. g. special verbal means in the natural language). F'' consists of formats which have been created artificially in order to be used effectively by computers. That is why the transformation $r_1(d, f_1) \rightarrow r_2(d, f_2)$ is usually performed easily by computers if $f_1, f_2 \in F''$ and with much more difficulties when $f_1 \in F'$ and $f_2 \in F''$. The paper [1] is devoted exactly to the last problem.

When we choose a format $f \in F''$ for every data type t we obtain the so-called internal language. Another element of the internal language is the set of relations between the data values. The check for correct relations is implemented by algorithms manipulating the corresponding data representations.

Now we can reformulate the problem in the defined terms. The first subtask, data extraction, is intended to find out whether an eventual data item (defined by its meaning) is available in the text and if so to determine its representation. The second subtask, data translation, performs the transformation $r_1(d, f_1) \rightarrow r_2(d, f_2)$, where d is any data item from the text, and f_1, f_2 are the formats of its data type in the text and in the internal language, respectively.

3. Description of the text classes. The formal description of the class of the specialized texts that are to be analyzed is the central point of the problem formulated. What we are interested in any text is data. That is why, first we have to determine the data items which may appear in the corresponding class. Second, the set of formats to be used in order to represent the data values in the text has to be determined for each data type. Third, there exists a piece of explanatory information intended to determine the data meanings and locations (representations) in the text for each data item and some groups of data items. Usually this information is expressed by phrases. For instance, the phrase "... will be, held in ... from ..." clearly determines the three data items (their meanings and locations) available in this sentence. The Backus metalanguage is a convenient tool for phrase description (an extension of it is proposed in [1]). Fourth, all the texts in a given class have a certain structure which consists of relations between the data items or more precisely between their meanings. Data is grouped by means of these relations in such a way that in each group one of the following properties might hold:

— data items should appear in a fixed order (e. g. in any recipe the necessary products precede the data about the way of preparing the meal);

— existense of one or more data items influences the existence of other data items (e. g. in case of an address we have either a city or a village, i. e. those items cannot exist simultaneously);

— no data item of one group may appear between two data items of another group (e. g. in any announcement for scientific event all the information about the way in which communications presented have to be formed is contained in a particular paragraph and is never mixed with any other information, for instance information concerning the financial conditions of participation).

The relations mentioned can exist between the groups of data as well. This property determines the hierarchical text structure which might be represented by one or more trees [2]. Most of these relations observe the traditions accepted. Finally, the two-dimensional representation of the text on the sheet carries additional information which could be used in the process of analysis. Since all the methods of analysis are based on string processing it is necessary to transform the text into a string with no loss of the additional information.

Summarizing the description of a text class we see that the following elements have to be defined:

a) the data items which may occur in any text of the class $d_1(m, t)$, $d_2(m, t), \ldots, d_n(m, t)$;

b) the set F of the possible formats used for representing the values of each data type;

c) a set of phrases $P=\{p_1, p_2, \ldots, p_m\}$, occurring in the texts of the class each phrase contains data and explanatory information;

d) text structure represented by one or more trees with special type of links between the successors of every node $l_i=l_i(d_1(m), d_2(m), \ldots, d_k(m))$, $i=1, \ldots, s$;

e) string representation of a text by preserving the additional information about the two-dimensional text arrangement on the sheet: $T \leftrightarrow S$ where S is a string of characters and T is the text.

4. Automatic text analysis. Text analysis can be presented from two points of view — at an abstract level as a composition of functions which transform the text into its internal representation and as a process (described by the corresponding program tools) implementing this transformation. The scheme according to which the described modules work is presented in Fig. 2.

The first transformation is the text linearization ($T \rightarrow S$). It consists of recognizing the sequential text units, such as paragraphs, titles and other special features of the text arrangement. The algorithms for this transformaion vary from simple text editing to complicated image analysis.

Then a lexical analysis is performed. It implements the mapping $S \leftarrow L$, where L is a string of lexemes. The dictionary containing words and word groups plays a central role in the process of scanning. That transformation has been studied theoretically and effective program tools have been developed in the field of translation techniques.

The syntactical analysis is an essential part of the data extraction subtask. It looks for phrases from P. The phrase searching can be illustrated by the pattern matching in Snobol. The explanatory information corresponds to the constant element in the pattern and the data — to the variables in the pattern, respectively. The unrecognized substrings of L are skipped as insignificant. As a result a set of interpretations (it may be empty as well) is obtained. Any interpretation consists of the data (defined through their meanings and representations) extracted by means of the phrase parsing process. This transformation leads to:

$$L \rightarrow \{I_1, I_2, \ldots, I_l\},$$

where $I_k=(d_1^k(m, r), d_2^k(m, r), \ldots \; d_q^k(\mathrm{m}, \mathrm{r}))$ is an interpretation of the text.

The fact that more than one interpretation is possible is due to the lack of strict rules for text formation. The existence of an empty set of interpretations may be due to some errors in the text or to undefined parts of the text class as well.

A context analysis (CA) is used to reduce the set of interpretations. Actually CA works together with the parser but for clarity it has been described here as a separate

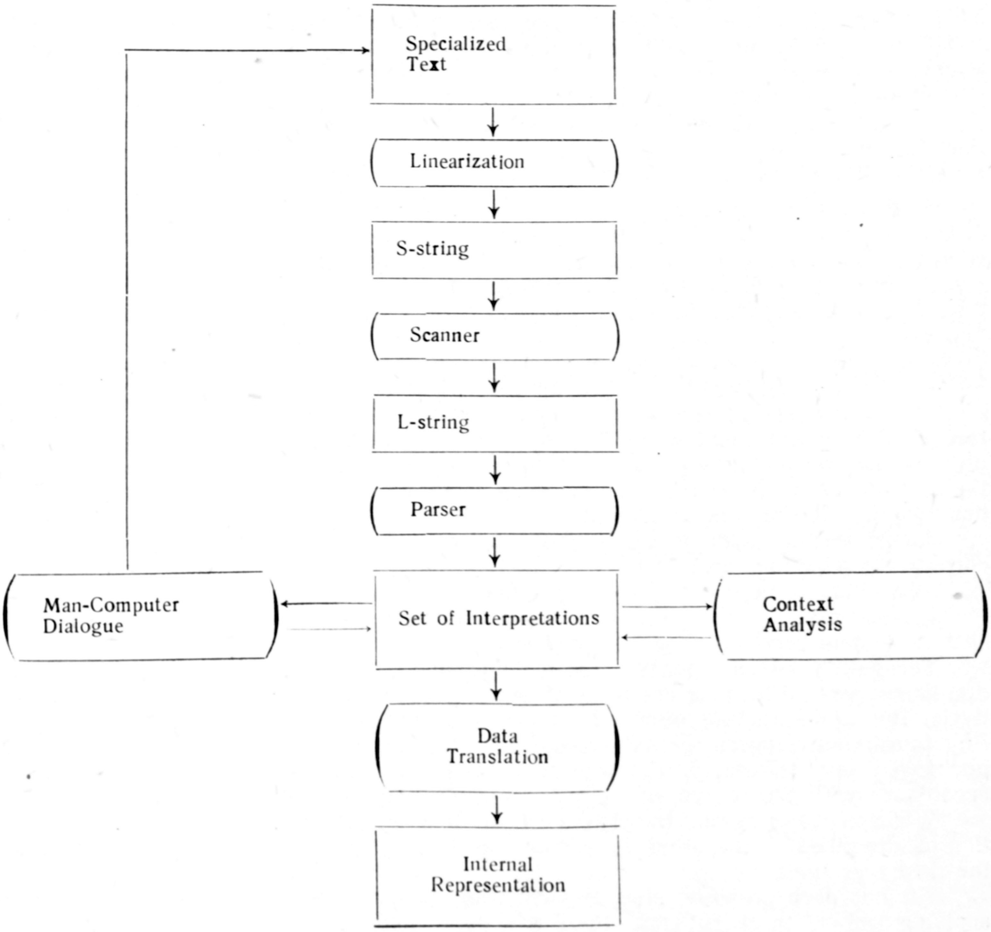


Fig. 2. Functional scheme for text analysis

stage of the whole process. CA exploits the relations described in the previous section. For each interpretation it checks whether the necessary trees could be built and if so it accepts that interpretation, otherwise — rejects it. A more detailed information about CA is available in [2].

The next stage, a man-computer dialogue (MCD), is optional. It is invoked if one of the following cases occurs:

— the set of interpretations is empty;

— the set of interpretations contains more than one element;

— the set of interpretations contains one element which is wrong.

The user could call the MCD in order to influence the analysis in one of the following ways:

— making corrections in the original text;

— choosing the right interpretation from the set of interpretations;

— determining the data items (their meanings and representations) in the text.

The data translation subtask is rather independent of the text analysis subtask. Each data translator is developed in the frame of a certain data type and as was said in section 2 it implements the function $r_1(d, f_1) \to r_2(d, f_2)$.

Since the data values from a given type can be represented in the text by different formats, the data translator has to determine first the format and then to transform the data value into the format of the internal language.

5. Discussion. An experimental system working in accordance with the scheme described was developed. Its modules have to be tuned up to a certain class of specialized texts. The essence of that tuning is the formal description of the class. This requires a careful study of the class. A reasonable question is who is to study and describe formally the text class. The answers may vary in a large scope between the following two extreme points of view:

— this work should be the only occupation of men of a new profession;

— this work should be fulfilled by the end-users among other things.

Those two approaches would require proper tools and techniques for the system tuning. The second stand would lead to more elaborated means of supporting the simpler and natural text descriptions typical of the end-users. The practical application of the first attitude is possible even now since it allows the text description to be carried out by using the existing formalisms.

An interesting analogy can be made between data and program translation. The corresponding notions are:

data value	— algorithm
format	— program language
data representation	— program.

The notions of an algorithm and a data value present two abstract entities. Program languages and formats are tools of representing algorithms and data values, respectively. The representations obtained are programs and data representations, respectively Any translation (program or data) keeps the abstract entity (algorithm or data value) unchanged and transforms the representations (programs or data representations) in accordance with the source and target languages (program languages or formats).

While in the program translation we know a priori the language, in data translation as described in the previous section the format has to be obtained by analyzing the data representation.

The approach proposed aims at facilitating people's use of computers. It is worth applying only if in every class there is a great number of texts which are already available on a computer readable media (e. g. to facilitate their updating by a text processing machine) or are entered directly by optical character reader. In the last case the method allows to increase the reader's precision on account of the additional information about the way in which the specialized texts are formed, i. e. the mechanism of CA aided by MCD can effectively be applied to error detection and correction.

6. Conclusion. The problem of data input has many aspects. In the paper propozed the difficult but feasible subproblem, extraction of data dispersed in a specialized text, is considered. The main aspects of the solution are:

— definition of the notion of a "specialized text";

— creation of a suitable data model and the place of the internal language in that model;

— formal description of a class of specialized texts including the possibilities for existence of undescribed parts of it (which would be considered insignificant);

— studying the concept of a data type and the different formats of representing the data values belonging to a certain data type.

Many questions arise when discussing the problems under consideration. A great deal of research is required in order to study the existing classes of specialized texts and to classify the data types and the formats of representing their values. In view of the practical application of the method much experimental work should be done. The successful development of that technique is feasible only through the collaboration between the specialists in various fields (applications, translation techniques, data specification, information systems, artificial intelligence, hardware).

REFERENCES

1. П. Барнев, Ст. Керпеджиев. Подход к вводу дэнных в свободном формате, примененный для дат и почтовых адресов. — В: Математика и математическо образование (13-та пролетна конф. на СМБ). С., 1984, 244—255.

2. Ст. Керпеджиев. Использование контекстной информации для распознавания составных, объектов. — В: Математика и математическо образование (14-та пролетна конф. на СМБ), С. 1985, 435—441.

Centre for Mathematics and Mechanics
Sofia 1090 P. O. BOX 373

Received 18. 10. 1985
Revised 20. 10. 1986.