Serdica
Bulgariacae mathematicae publicationes

Сердика
Българско математическо списание

SERDICA — Bulgaricae
mathematicae publicationes
18 (1992) 206-214

AUTOMATIC CONVERSION OF ENCYCLOPEDIA ENTRIES INTO A HYPERTEXT

M. DOBREVA, ST. KERPEDJIEV

ABSTRACT. The fundamental problem of the creation of real hypertext systems has been tackled in the case of converting *Ancient and Classical Literature* encyclopedia into a hypertext. We apply pattern-matching techniques for automatic extraction of data from documents in order to specify the nodes and links of the hypertext. Our approach relies on the use of a conventional DBMS for organizing the hypertext and emulating the navigation. An experiment designed to evaluate the method of data extraction shows that 97% of the data items are extracted correctly.

1. Introduction. The increasing interest in the field of hypertext systems raises some fundamental problems. One of them is: *How to create a real hypertext system?* This question could be further specified with respect to the subject of creation – a human (expert in the subject domain or computer scientist), a computer (i.e. the creation is performed automatically), a human supported by a computer, or a computer supported by a human. The answer depends on various factors such as type of the original material, requirements on the structure of the hypertext, available techniques and tools for conversion of the text into hypertext, etc.

We suggest a method of converting text into hypertext based on pattern-matching techniques and conventional DBMSs. According to this method, a human expert makes a conceptual analysis of a sample text, extracts the text portions that identify the nodes and the links, and describes those portions formally as text patterns. Then the computer tuned by the patterns converts automatically an arbitrary text from the same class into a hypertext or a part of the hypertext. The method is applicable to structured original texts only and has certain advantages - simplicity, efficiency and high quality.

This method has been applied in the case of conversion of *Ancient and Classical Literature* encyclopedia into a hypertext. Experiments designed to evaluate the quality of the conversion have been performed and analyzed.

The paper is organized as follows: Some basic concepts are briefly introduced in Section 2. In Section 3 we discuss the possible strategies for creating hypertexts. In Section 4 our method is treated. In Section 5 the particular case of an encyclopedia hypertext is considered in details.

2. Hypertext. There are different views on the essence of hypertext. A comprehensive survey of hypertext systems can be found in [2]. Our conception about hypertext is similar to that given in [10], namely hypertext systems are "valuable tools for creation, (re-)structuring and presenting information bases". There are three characteristic points in this definition:

- The hypertext is a *tool.* The understanding of a hypertext as a special kind of text whose parts are interconnected implies that all books are hypertexts, because they contain links between their parts.
- Hypertexts are used for *creation*, *(re)-structuring* and *presenting* something. We could add the function of *storing* to this list.
- The objects maintained and handled by the hypertext are parts of an *information base.* In the general case, they may be photos, graphics, texts, etc., but in a pure hypertext system, they should be text portions.

For the sake of convenience we call the information base organized and handled by the hypertext system *hypertext* too. These two meanings will be distinguished by the context.

According to this and most of the other definitions, the basic elements of a hypertext are *nodes* and *links.* The nodes contain pieces of information (texts, graphics). The links provide direct access from one node to another. They inspire power in the tool, because the effectiveness and the efficiency of the hypertext depends heavily on the interconnections between its nodes.

The nodes and links form the hypertext *web.* The access to a particular piece of information usually requires passing through some nodes following a certain path. The problem of selecting the best route is known as *navigation.*

The problem of hypertext creation consists of determining the set of nodes, their contents and the links between them.

3. Strategies for hypertext creation. The strategies described here pertain to the case of hypertext creation when the information base is originally available as a linear text (e.g. in the form of a book). We do not consider the case of hypertext authoring systems.

The first and most often discussed strategy for hypertext creation consists in choosing the nodes and making the links with human participation in the process of using the hypertext. This strategy is reasonable when the nodes and the links of the hypertext are non-typical and their extraction from the text requires a human mental effort. The importance of correctly selecting text fragments for nodes was discussed in [8].

Another strategy with significant human participation consists in using *mark-up languages* [7]. It differs from the first one in the mode of human-computer interaction.

In the first case, a special user interface should be provided for interactively selecting the nodes and making the links. In the second case, the user marks up the original text to show where a node or a link should be created, and the system performs the task in a batch mode.

A third strategy with two variations relies on *indexing* with a controlled vocabulary and *classification* with an uncontrolled vocabulary. Frisse employed a list of keywords prepared by an expert to index the parts of a medical handbook [5], and Coombs proposed full-text searching techniques for finding words of a special interest for the user and linking the text parts properly [3].

A fourth strategy given in the literature is to create those links that are *explicitly* used in the book, e.g. list of contents, indices, cross-references, footnotes, etc. [9].

The strategy we propose is based on using *pattern-matching* techniques to find out the text portions that can be used as nodes automatically and to extract links, both implicit and explicit, between the nodes. Compared to the four strategies discussed above, ours is the nearest to the third and fourth ones. It requires preliminary conceptual analysis by an expert and employs some links explicitly expressed in the original text.

4. Method. According to our method the conversion of text into hypertext is performed in two stages – text analysis and hypertext generation.

The first stage ends with the specification of each node of the hypertext with respect to content (the portion of the initial text that constitutes the node) and references (keywords and phrases that indicate the links and can be used for navigation). The hypertext generation consists of designing the DB scheme, filling the DB with textual nodes and data extracted in the first stage, and cross-connecting the nodes.

4.1. Text analysis. For the first stage we employed a system for text analysis developed in the *Institute of Mathematics* in Sofia and described in [6].

The methodology of using this system requires passing through three steps: conceptual analysis, formal description and computer analysis.

The conceptual analysis of the texts from the subject domain is intended to define the text class. It is carried out by a human who reveals the information and textual structures of the texts. As a result, two sets emerge – a set of data types constituting the information structure and a set of patterns forming the textual structure. Furthermore, some of the patterns, called data patterns, define the textual values of data types, thus implementing the relationship between the textual and the information structures of the text class. Another group of rules takes care of the order of occurrence of data items (resp. patterns) in the texts. It can minimize the time for text analysis and may protect the system from obtaining incorrect results.

The system for text analysis provides a formal language for description of the text class at three levels:

- Description of the types of the data items to be extracted from the text. Three

simple and three compound data types can be used – *integer, string, enumeration* and *record, set, list,* respectively.

- Description of the patterns. The patterns consist of indicators of four types – *word class, word, numeral* and *punctuation mark*, and are composed by means of five operators: *concatenation, alternation, negation, optional element* and *loop.*

- Description of the order of occurrence of data items in the text. The rules prescribe which patterns are to be activated as a result of the successful matching of a given pattern.

The formal description tunes the system for text analysis to process the texts accordingly. The result of the analysis is a list of data items extracted automatically from the original text.

4.2. Hypertext generation. The generation is carried out in three steps: design of the DB scheme, filling the DB and making the links.

The design of the DB scheme is based on the results of the conceptual analysis. The relations included in the DB are of three types: *t1)* this relation contains all attributes with a restricted number of occurrences in an entry; *t2)* a relation of this type contains a key combination of attributes and an attribute with unrestricted number of occurrences in one entry; *t3)* the relations of this type contain meta-information.

Each node, regarded as a text, is disposed in the hypertext store. The list of data items attached to the node is scanned and each data item is added to the relation corresponding to its type.

In fact no explicit links exist in the system. We use the term "create a link" to express the transition from one node to another through a conceptually existing link. Before making a link, the text of the current node is shown on the screen with displayed data fields. The user can reach any field and can activate it. The activation triggers the system to search the corresponding relation for entries having values identical with those of the activated data field. The list of the selected entries is shown to the user, who by picking the name of the desired entry completes the transition along the link.

Such a method of creating links between nodes minimizes the amount of data stored in the DB, assures tolerable access time and allows convenient user interface for the navigation.

5. Case study. *Ancient and Classical Literature* encyclopedia in Bulgarian [1] was used as a testbed for the method described in Section 4. We will describe concisely the results obtained at each stage of the conversion.

The encyclopedia entries contain information about the lives and the works of authors from Ancient Greece and Rome. An excerpt from an entry from a similar encyclopedia in English [4] is given below to illustrate the original material.

Data description	Identifier	Number of items in one entry
author's name in Bulgarian	NAME	1 - 2
author's name in Latin	LATNAME	1 - 2
town where the author lived	TOWN	0 - 1
region where the author lived	REGION	0 - 1
century in which the author lived	CENTURY	0 - 2
author's birth year	BYEAR	0 - 2
author's death year	DYEAR	0 - 2
origin	ORIGIN	1
genre(s) of author's works	GENRE	1 - 4
cross-references	REFERENCE	unrestricted
title of work in Bulgarian	OPUSBUL	unrestricted
title of work in Latin	OPUSLAT	unrestricted

Table 1: Data types in the *Ancient and Classical Literature* encyclopedia

Alcaeus, of Mythilene in Lesbos, the earliest of the Aeolian lyric poets, b.c. 620 B.C. In the war between the Athenians and Mythilenaeans for the possession of Sygeum (606 B.C.) he was disgraced by leaving his arms on the field of battle. ...

5.1. Conceptual analysis. The data extracted from the texts are described in table 1. They are of various types – mostly strings, but GENRE and ORIGIN, for example, are of enumerated types. Two names for an author may appear in an entry, due to differences between the various sources of information about his life.

The data set extracted from one entry can be divided into two subsets. The first one includes personal data, which do not relate to other authors (e.g. OPUSBUL). The other group includes data that can be used for creating links. The links are of several semantic types as described below:

Living in the same town. Connects all authors who lived in the same town. For instance, the Alcaeus entry should be linked with all the entries about authors living in Mythilene.

Living in the same region. Connects all the entries about the authors who lived in the same region (the island Lesbos, for the Alcaeus entry).

Living in the same time. Connects the entries about the authors who lived in the same century.

Writing in the same genre. Connects the entries about the authors who wrote in the same genre.

Cross-reference. Connects two encyclopedia entries, the first one containing an explicit reference to the second one. Its semantics may vary.

Next entry. It provides the alphabetical order of the entries. This relation determines the sequential nature of the original encyclopedia and its implementation in the hypertext still allows the users to explore the material in the traditional manner.

The last two types of links are the only ones that provide direct access to the original encyclopedia.

The textual structure of the encyclopedia entries is described by means of 25 patterns. Nine word classes have been defined and used in the pattern creation. Some other typical indicators involved in the patterns are: words beginning with capital letters (for proper names, titles); specific punctuation marks and symbols (e.g. parentheses for the names in Latin); numerals (for years and centuries).

The relation of partial order between the occurrences of data items in an entry is described by means of the oriented graph in figure 1.

5.2. Formal description. An excerpt from the formal description is given below to illustrate the means of expression of the language. It includes the patterns describing the occurrence of the data items NAME and REFERENCE.

```
*** DATA DEFINITION
NAME IS STRING;
REFERENCE IS STRING;
*** PATTERN DEFINITION
DP NAMP(NAME)=(WO CYR FC):1:2 [{'and','or'} (WO CYR FC):1:2];
DP REFERENCEP(REFERENCE) = (WO CYR FC):1:2;
PP REFERENCEPP = REFERENCEP '*';
*** PATTERN LINK DEFINITION
->NAMP;
NAMP -> LATNAMP;
REFERENCEPP -> REFERENCEPP, OPUSBULP;
```

In section DATA DEFINITION, the two data types are defined as strings. Section PATTERN DEFINITION consists of three statements defining two data patterns (DP) and one phrasal pattern (PP). The data type whose textual value is defined by a data pattern is given in parentheses just after the pattern identifier. The pattern bodies (on the right-hand sides of the statements) contain indicators such as literals, words specifications (e.g. WO CYR FC meaning a Cyrillic word beginning with a capital letter) composed by operators such as concatenation, alternation denoted $\{A, B\}$ and meaning "A or B", etc. Section PATTERN LINK DEFINITION consists of statements, each with two sides separated by the '– >' symbol. The left-hand side of a statement contains the pattern-predecessor and the right-hand side of the pattern(s)-successor(s).

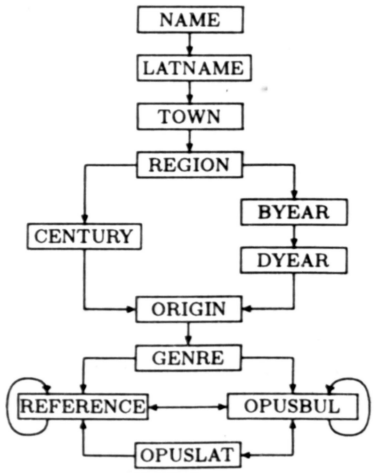

Figure 1: Data order.

5.3. Computer text analysis. The text analysis system tuned by the formal description of the entries segments a portion of the original encyclopedia into entries and creates a list of extracted data items for each entry.

In order to evaluate the performance of the system, we made an experiment with a portion consisting of 73 entries (the average number of characters in an entry is 1100). The extracted data items were analyzed for relevance and correctness (the results are shown in table 2). All incorrectly extracted data were titles of works in Bulgarian. The reason is the occurrence of titles of works by other authors in the analyzed entries. This fact allows us to narrow down the set of potentially erroneous extracted data that have to be checked by a human.

Criterion	Value
Number of relevant data (R)	948
Number of extracted data (Q)	968
Number of incorrectly extracted data $\|Q - (R \cap Q)\|$	20
Precision $\|(R \cap Q)\|/\|(Q \cup R)\|$	97%

Table 2: Computer analysis results

5.4. DB scheme design. The DB scheme consists of 6 relations. The first one is of type *t1* and includes the attributes NAME, LATNAME, TOWN, REGION, CENTURY, BYEAR, DYEAR, ORIGIN and GENRE. The attributes REFERENCE, OPUSBUL, OPUSLAT correspond to three other relations of type *t2*. The fifth rela-

tion includes information about the towns situated in each region. The last relation represents the hierarchical structure of the common literary forms (e.g. drama) and their branches (e.g. tragedy, comedy). The last two relations are of type *t3*.

5.5. Filling the DB. The corpus of analyzed entries was used for generating the hypertext. As a result 73 nodes were created. The potential number of links of each type for the present hypertext content is given in table 3. The great number of links of type *Living in the same time* is due to the fact that most of the authors considered in our experiment lived in the same century.

Type of the link	Number of links between pairs of nodes
Living in the same town	24
Living in the same place	18
Living in the same time	1348
Writing in the same genre	1132
References	225
Next entry	72

Table 3: Number of links

6. Conclusion. Hypertext systems have certain advantages over traditional texts. They facilitate the exploration of various problems that require studying text portions linked in a certain way. An example of such a question is: "What characterizes the authors who lived in Lesbos?". If we use a traditional encyclopedia, we should browse through the whole text in order to extract the necessary information. Conversely, hypertext allows us to get a direct access to exactly those entries that are relevant to this question.

We suggest a specific method for converting text into hypertext. The method consists of conceptual analysis of the text class, formalization of its textual and information stuctures, automatic data extraction, and automatic generation of the hypertext based on the data extracted. This procedure has been applied to *Ancient and Classical Literature* encyclopedia. The results show that 97% of the data items are extracted correctly, which requires little user intervention in the correction of the wrongly extracted data.

This study raised a number of problems that need further investigation:

- To what extent can such a method be applied to other types of original texts? Potential objects of research may be other encyclopedias (Mythological, Historical, etc.).

- What types of queries can conveniently be processed if a hypertext like this one is used? We know that no hypertext can be a panacea for the information retrieval problem, yet we should know more precisely how worth our system is.

- How does the number of links influence the performance of the system? This question concerns both the efficiency of the system and its effectiveness.

In order to explore the questions we put, a complete hypertext version of the encyclopedia must be produced so that real large-scale experiments could be performed.

REFERENCES

[1] BOGDANOV, B., A. NIKOLOVA, editors. Ancient and Classical Literature. Narodna kultura, Sofia, 1988 (in Bulgarian).

[2] CONKLIN, J. A Survey of Hypertext. MCC Technical Report STP-356-86, October 23, 1986.

[3] COOMBS, J. Hypertext, Full Text and Automatic Linking. SIGIR '90 Proceedings (ed. by J. L. Vidick), ACM, Brussels, September 1990, 83-98.

[4] Everyman's Smaller Classical Dictionary. London, 1956.

[5] FRISSE, M. From Text to Hypertext. *Byte*, **12** (1988) 247-253.

[6] KERPEDJIEV, S. Automatic Extraction of Information Structures from Documents. ICDAR '91 Proceedings. AFCET, Saint-Malo, France, September - October 1991 (in print).

[7] MCKNIGHT, C., A. DILLON, J. RICHARDSON. Hypertext in context, Cambridge University Press, 1991.

[8] RAYMOND, D., F. TOMPA. Hypertext and the Oxford English dictionary. *Comm. of the ACM*, **31** (1988) 871-879.

[9] REMDE, J., L. GOMEZ, T. LANDAUER. SuperBook: an automatic tool for information exploration – hypertext? Hypertext '87 Proceedings, ACM, Chapel Hill, NC, November 1987, 175-187.

[10] SCHUTT, H., N. STREITZ. Hyper base: A hypermedia engine based on a relational database management system. Arbeitspapiere der GMD 469, Darmstadt, July 1990.

Institute of Mathematics
Bulgarian Academy of Sciences
Acad. G.Bonchev str., bl. 8
1113 Sofia,
BULGARIA

Received 28.12.1991