On the Functionality of Text-based Management Systems for Literary Research

George Mitrevski
Department of Foreign Languages
Auburn University
8030 Haley Center
Auburn, AL 36849

Introduction

The proliferation of microcomputers among non-traditional users, such as scholars in the Humanities, has made it cost-effective for many software companies to develop low priced database systems designed to facilitate more efficient manipulation of textual data. Although not as powerful as those found on mainframes, these systems can simplify greatly many tasks of the literary scholar. Obviously, none of the presently available programs answer all the needs of every scholar in the field. The reasons are many, but the two most obvious are the following: (1) programs lack sophistication because most software designers, generally trained to construct systems for the business and scientific communities, have only rudimentary understanding of the tasks involved in conducting literary research; and (2) there is a misconception among majority of literary scholars about the capabilities of computers and software. Computers are most often perceived by them as "universal" problem solvers, and they feel frustrated when the machine and software cannot solve their own particular problem. In many of these instances, the problem cannot be solved either because the user does not know how to formulate it in a way that it becomes comprehensible to the problem-solving system, or because the solution is beyond the capabilities of the system.

These overriding concerns are exemplified in specific problems encountered when using standard data-base management systems for managing textual data, the conceptual problems of designing systems that can perform the most essential tasks in conducting literary research, and finally, examples of the approach used by three text-based management systems designed specifically for manipulating narrative texts.

Data- vs. Text-based Management

The notion of text-based management system is rather new. The essential difference between data-base and text-based systems is in the type of data they are designed to manage. The former are normally designed to handle numerical data very efficiently. They can manipulate textual data as well, but only when the 'units' of data are not very large. Text-based systems, on the other hand, are designed specifically for manipulating textual data presented in narrative form. Mathematical capabilities of such systems are limited to simple arithmetic operations. The main difference, then, is functional: it is based on their efficiency in handling data of both types.

If they are to function very efficiently, traditional data-base management systems require strict classification (typing) of the data (both textual and numerical). When working with such systems, all data that cannot be classified is disregarded as inessential; it can be accessed, but it cannot be manipulated. The units of data in traditional data-base systems are quite small. Classifying data usually implies identifying the functional superstructure of the 'data-mass', where every item is identified to belong in one or another category within the superstructure. What usually happens in reality is that the user first creates the superstructure, and then selects appropriate data to be filled in. For example, if one wanted to create a name and address database, one would create the structure, identify the categories of data needed, and then out of the mass of data available about an individual, he selects only those items which can be included in the structure. Elements of data are either selected or eliminated. All eliminated elements have zero informational value.

The data selection process endows every selected element with a 'class identity', which is traditionally referred to as the variable, or field. Every element in the data is one instance, or value, of the variable. It is important to note here that once a unit of data is assigned to a variable, it is stripped of all nuances and ambiguities in meaning. Only the meaning, or range of meanings assigned by the variable, or category name, are acceptable. The number of categories into which the data is organized must be finite. This is why it is possible to classify an enormous mass of data.

A system designed to manipulate the above type of data cannot be very productive in working with narrative texts of the sort encountered in literary texts and commentaries, because such data does not lend itself to easy classification. The main reason for this is because it is difficult to identify the exact units in the text which contain information. Information in such texts is nested in the various linguistic structures: word, sentence, paragraph, context, etc. Moreover, information there does not reside in a closed system, but is the result of an active encounter between the text and its recipient - the reader. Thus, while the linguistic structures remain the same, the information they carry may change with each encounter of the text. This semantic instability of the text is what prevents categorization of narrative textual data.

Managing Narrative Textual Data

Manipulation of data in the business environment generally entails selecting, structuring and ordering the data, and performing arithmetical operations on numerical variables. Due to the relatively small size of textual data contained in each variable, and the fact that every unit in the mass of data is classifiable, it is possible to extract from the data-base a great deal of information with very little difficulty. Note, however, that the rigid structure of the data prevents one from performing an unlimited number of operations on it. Although the operations may be numerous, they are nevertheless finite.

In many instances these traditional data-base systems can be used quite well for certain types of research involving narrative textual data. On the most basic level, they can organize narrative text data according to the specific characteristics of the linguistic units - the words. There are numerous programs that can read from a file of unstructured textual data, create a sorted index of all the words and perform statistical analysis, such as percentage and frequency counting, which tabulates the number of times that the values of particular variables occur (e.g., SAS: Sohl, 1979). There are limitations, however, in using a word index for literary analysis, because of the limited amount of information that can be extracted from it. Unfortunately, most current text-based management systems are limited to operations on texts via an index.

Traditional data-base systems can be more practical in literary research for analyzing the grammatical structure of the texts. Because of the limited number of grammatical categories in a language, it is easy to provide a label for each word and to group all words according to such categories. Textual manipulation of this sort can provide for the scholar statistical information about prevalent usages of certain word categories over others. This information can be contrasted with various thematic structures. This approach can be impractical: one would have to categorize every individual word, and in the instance of very large texts, the process would be quite time consuming. The other possibility is to automatize the process by using a program which would read a word from the text to be analyzed, it would search for the same word in its dictionary, and it would automatically mark the word in the text with an appropriate category marker.

By far the greatest use of computers in literary analysis has been in creating concordances of texts. Concordance programs are very similar to indexing programs, except they can provide more detailed information about the environment (context) in which a word is situated.

The types of textual analyses described above are certainly extremely valuable in understanding literary works, and the contribution of computers and software toward that end cannot be overemphasized. However, this approach to literary texts yields little information in comparison with the information.contained on the semantic and thematic levels. The problems of devising management system capable of accessing this information directly from the text, or at least that are capable of directing the user to the place in the text where the information is located, open myriad questions.

For a thorough understanding of the problem, first one would need to define the term 'information' and identify the processes through which it manifests itself to the perceiver. Such a definition may be limited to instances when information is manifested in numerical and verbal forms. The simplest definition of information is that it is a fact which is coded in a signifying system. In order for information transfer between two systems to take place, the code must be shared by both systems. Information in a standard data-base is coded through language. When a mass of information is categorized into a data-base, it acquires specific characteristics. The signs (words, digits) in a data-base contain information only when they are perceived as values of a particular variable. The variable imparts the information value on the sign. Outside of this relationship the sign has zero informational value. Thus, the signs 'Mitrevski' and '36' are meaningless unless they are attached to a variable, such as 'NAME' and 'AGE'. When the sign '36' enters into a signifying relationship with the variable 'NAME', as ridiculous as it may sound, the information perceived is that the value '36' is the name of some entity. Consider, for example, the case of an acquaintance who owns a small publishing company named Slavica Inc., who not too long ago received a letter requesting donation for some cause. The letter was addressed to Slavica Inc., and the greeting line read "Dear Mr. Inc.". Apparently the system was set up to read the first item of the database as the first name and the second as the last name. Although the system can be set up to filter out certain values as improper for a particular variable, it cannot do this exhaustively because it cannot make inferences about every possible absurdity in the real world. While the juxtaposition of 'Mr.' and 'Inc.' in the world of the data-base system is quite comprehensible, the same juxtaposition in the real world is semantically incongruous.

Consider another example. The variable 'HOURS WORKED' in a traditional data-base will accept the value '37.5', but not a value such as "Last week I worked only about thirty-seven and a half hours." The reason is not necessarily because the value is represented in narrative form, but because the value is vague, and therefore inexpressible in a form acceptable to the system. While it is possible to devise a system that can read the narrative phrase, look for verbal expressions of the value needed and translate the value into numerical code, in the process, however, the procedure will eliminate all other verbal signs as 'noise'. But, it is this 'noise' that contains information that makes the value '37.5' vague, i.e., to the adverb 'about'. It is precisely because of problems such as these that traditional data-base systems need a rigid system of organizing data.

The problem of organizing textual narrative data arises from the difficulty in identifying the specific units of data that contain information. The questions one may ask when dealing with literary texts are, what exactly is the information contained in the data and where is it located? As noted earlier, the linguistic and grammatical units identifiable in narrative textual data have very little value independently. Information in such texts is contained in the concepts expressed when the words are combined into higher linguistic structures. Independently, words have 'denotative' meaning; the literary scholar however, is interested more in the 'connotative' meaning of words and expressions, when words mean something other than what they say (a distinction discussed by Barthes, 1967).

As noted before, in a conventional data-base the informational value of the word is established only in relation to a particular variable, when it is entered as its value. Outside of the context of the data-base, the word 'Mitrevski', for example, may contain limitless amount of information, especially for a person who is familiar with the linguistic structure of the word and with the reality to which the word is attached. The word indicates that there is a reference to a human being. The last three letters indicate that he is of Slavic nationality, and that the subject is a male. In a data-base variable, however, only one, unambiguous reference can be attached to the word; all other references must be stated explicitly in additional variables, or subordinated meaning can be coded in the tag structure of the variable.

Related to this problem, is that of dealing with grammatical elements like pronouns and certain adverbs (such as 'here', 'there', 'yesterday', etc.) which have no precise meaning; they may carry different information with every occurrence. The meaning of each of these 'shifters' (Eco, 1979) is determined by the context in which they occur. If, for example, one looked in a text for all references to a literary character, the program would not be able to detect instances where the name of the character was substituted by a pronoun. One possible, but time-consuming solution is to go through the text and attach the intended reference for each of these 'shifters'

One can now understand the enormity and complexity of a data-base if all the information contained in every word in the text must be stated explicitly. The interest of the literary scholar is in comprehending the semantic ambiguities of expressions and the multitude of references that a single expression may have; this constitutes a major part of his scholarly research. This is when frustration with computer applications in research begins.

The problem of extracting information from narrative textual data increases manifold when considering units of meaning larger than a word, such as sentences and paragraphs, and even larger blocks. This problem was also recognized by scholars working with theories of narrative texts. Narrative, according to Chatman (1978), "as the product of a fixed number of statements, can never be totally 'complete,' in the way that a photographic reproduction is, since the number of plausible intermediate actions or properties is virtually infinite." A program can identify each of these grammatical and syntactic units easily by searching for specific markers. But, other than to identify them as such, it cannot extract additional information about any outside reality they may represent. Information contained in a sentence is greater than information contained in the sum of all the words which constitute it. As example, consider the following sentence in an imaginary literary text: "Her brother died, and she cried all week". Each individual word here has meaning, or contains information, but there is also additional information to be extracted from the sentence as a unit, i.e., information constituted in the juxtaposition of the two phrases. This information is that perhaps she loved her brother, his death is the cause of her crying, and she was sad and depressed. The context in which the sentence is situated may also attach additional meanings; it may even provide information that contradicts inferences made earlier. Further in the text one may read: "The next day she told everyone that she was faking her crying, and that her brother was not dead." It would be futile in this instance to search for information about a person's reaction to the death of a brother without taking the context into consideration. The problem of extracting information from textual data is obviously exacerbated manyfold in approaching the higher organizational levels of the textual mass.

Extracting Information from Textual Data

Data is empty of all information until it is subjected through a reading by an interpretative system. The quality of being an "information carrier" is imposed on the data by the system. In a traditional data-base, data acquires information through its identification with and attachment to a variable. The system cannot extract from the data any information outside this relationship. Therefore, the system can manage narrative texts only to the extent that such relationships can be formulated between the textual data and a variable. In establishing this relationship, the system sets stringent rules for the 'reading' of the text; multiple readings cannot be accommodated, because the information that the data carries is reduced to a finite value.

Reading literary texts in this manner is problematic because information in such texts does not reside in a closed system (variable-value), but in fact is conceived through the participation of a reader. The reader is not only a receiver of information, but acts as a participant in its creation as well. As Iser (1978, p. 107) has noted, "Reading is not a direct 'internalization', because it is not a one-way process .. [it is] a dynamic interaction between text and reader." The amount of information generated from the data will depend largely on the amount of information the human participant contributes in the reading process. Every reading of the text constitutes generation of additional information, which may be different from information received during previous readings. Successful 'transfer' of text to reader, according to Iser, "depends on the extent to which this text can activate the individual reader's faculties of perceiving and processing." Since every reader contributes unique experiences to the reading process, information that can be extracted from the text can never be stable; this is why it is impossible to identify what exactly is the information carried by the linguistic structure, and to label it as permanent.

Although, as Todorov (quoted in Chatman [1978], p. 17) has stated "Any work is .. its own best possible description: entirely immanent and exhaustive," sometimes it is necessary to make explicit the text's organization and some of its principal traits. Armed with this knowledge one can participate more actively in the reading process of the primary text, and consequently extract more valuable information from it. In fact, the most timely activity of the literary scholar is that of reading secondary text for information.

How can text-based management systems help in gaining this knowledge? Before answering, first one must dispel the fallacious notion that it is possible to design computer hardware and software capable of 'interpreting' literary texts (if 'interpretation' is understood as a subjective activity controlled largely by the reader). Electronic systems can be quite useful in extracting certain types of information from primary texts, as indicated by aforementioned several such suggestions of statistical studies of texts. Three possible approaches toward storing and accessing textual data for literary analysis, using as examples text-based management systems designed for microcomputers, are general purpose systems. There are probably hundreds of others designed for use on larger machines, but they are inaccessible to majority of users and are usually limited in their functionality because they are usually designed to perform only specific kinds of tasks.

To make the following discussion more comprehensible, imagine a disc full of articles on "Hamlet" in ASCII format, each is stored as a separate file. To write a paper on the play one needs some references on a specific topic, such as all the elements, both structural and thematic, that make the play a tragedy; how each character contributes toward the tragic action of the play. The next step is an attempt to find out if any of the authors in the volume make any references to this subject and where in each of the articles is this information contained.

From this point on, how much information is extracted from the articles depends largely on how much time is spent searching for it. The quickest and least time consuming approach in searching for the information is through an index of all the words in the files. The index can be useful only if it can point the place in the text where each indexed word occurs.

One of the shortcomings of this approach in searching for information, is that the subject may not be expressed necessarily in a vocabulary item, but may be constituted in a phrase, or a sentence. The next option would be to read the texts, as one would read a hard copy, take notes, mark particular places of interest and extract quotations. One can summarize, describe the text, and make explicit information that is contained in higher level grammatical structures. Description, according to Todorov is "a reasoned résumé; it must be done in such a way that the principal traits of the object are not omitted and indeed emerge even more evidently. Description is paraphrase that exhibits (rather than conceals) the logical principle of its own organization" (quoted in Chatman, 1978, p. 17). In the process of responding to the texts, notes are created as verbal metatexts. This activity is no less time consuming than reading hard copy. However, a text-based system can be very helpful in organizing the reader's responses to the primary and secondary texts. The advantage of this approach is much more apparent when working with a mass of data that consists primarily of reader's notes. The search for information here is much more productive than when searching indexed data, because extracted thematic information from the text is expressed using one's own vocabulary.

One very effective text-based system for storing research notes is *Notebook* II.[1] This program allows one to define a data-base with variable length records and fields, limited only by the space available on disk. One can easily import each article as a separate record, provide a field for the AUTHOR, TITLE and the full TEXT, and gain access to the full text of each article without any modifications. In order to keep the original article untouched, COMMENTS and KEYWORDS are stored in separate fields. Both the original texts and comments can be searched for occurrences of specific words and phrases, using Boolean logic. If a word processor is used, once a search is completed, all selected texts from a data-base are easily transferable into word processing files.

The greatest advantage of using such a text-based system is that it does not require much time to structure the data-base and import files. The data is presented to the reader in a form similar to familiar 3 x 5 or 5 x 8 cards. The system can select and access data, but it cannot manipulate it. Therefore, it is not recommended for working with primary texts, but solely for storing and accessing secondary sources..

A different approach toward storing and accessing textual data is provided by the text-oriented data-base system *askSam*.[2] This system is truly free-form; i.e., it does not require that data be classified and entered into fields, but it can be entered into a record as straight text, anywhere in a record. The maximum size of text in a record is 20 lines. To accommodate longer texts, a group of records can be linked to form a document of unlimited size. Information can be retrieved based on any combination of words, symbols, or 'wildcards', as well as logical expressions with AND, OR, NOT, including LOOP and nested IF..ELSE..END constructs. Numerical capabilities are quite sophisticated, including the functions SUM, COUNT, AVG, MAX, and MIN. After a query, askSam creates a file containing all the matching records.

While fields are optional in askSam, use of fields allows specific parts of records to be referenced. They can be added to the text any time after the text is entered. There is no formal definition of fields, which are referenced not by their position, but by their name. Therefore, askSam allows duplicate field names within a record. There are three types of fields available: (1) implied, such as $200.00 where the $ can be used as the field name for the purpose of record selection and other operations; (2) contextual, such as "Name George Mitrevski" where the word "Name" is used as a field name, and the text that follows up to the end of the line is the value; and (3) explicit, such as "Name [George Mitrevski] where "Name [" is the field name. Only the text that follows the [and ends with the] is the field value.

This text-base system could have been very practical in literary data analysis, had it not been for some major deficiencies in dealing with large texts. The system is not very efficient when it comes to storing records with large texts. Although it allows linking of records into documents in order to accommodate large texts, text does not scroll automatically from one record into another. If a record is full (20 lines), it is impossible to add additional text to it.

The advantage of working with non-fixed fields is that the user can select which parts of the text need referencing. A request for selecting particular records outputs only the values for the specified fields, rather than the entire record. In *Notebook II*, on the other hand, the entire record is output. Another problem with explicit fields in askSam is that the size of the field name and its value cannot be longer than a line. If the value is longer than a line, duplicate field names must be used on each line. This would require much work on the part of the user, and the duplication of field names leads to unnecessary loss of storage space. This text-based system is much more appropriate and functional for research in the social sciences, where data items are short, and the emphasis is on manipulating short, nonnarrative data.

The third, and by far the most sophisticated approach to storing and accessing narrative textual data is that offered by ***Text-Base***, which is part of the ***Nota Bene*** word processing system.[3] *Text-Base* can be used independently of the word processing system or in conjunction with it. There are several unique features about this system: e.g., there is no option for fields in Text-Base; the record is the smallest structural unit; and it offers flexibility in defining the boundaries of a record, or entry. The use of records in this system is the key to getting only specific parts of the text returned. One of the shortcomings about *Notebook II* is that, if an article was contained within a single record, successful search would return the entire text of the article regardless of size, but *Text-Base* offers many flexible possibilities in structuring the text in such a way that a query returns only the desired blocks of text.

Text-Base can view an entire document as a single record. Before the text can be searched, the system creates an index and a vocabulary file of all the words in the text. If all files are used, every one needs to be indexed into a master index and vocabulary file. This is the fastest, but not very productive way of searching the file for information. If units smaller than a file are used file can be broken into records by specifying their boundaries (markers) either by using predefined formats for marking ends of entries, or customized marks. Depending on the format of the text, the fastest way to format entries is to define the paragraph marker as the entry separator. If the paragraphs in texts were separated by a blank line, a format may be selected which treats a blank line as a separator. With practically no effort a search for information will yield only the appropriate paragraphs from each file.

All together there are eight predefined entry formats in *Text-Base*. In addition, the system allows the user to define formats according to one's needs. The most functional formats for literary research are Formats 5 through 8. The basic approach toward defining entries using these formats, is that they allow indexing only those parts of the text which are significant. This is accomplished by inserting control markers ('prompt markers' in *Text-Base*) at the beginning and end of the entry..Format 5 for example, treats each block as a separate record, and only text within the block is indexed. This format is useful when working with texts containing very large blocks that are irrelevant for the topic researched. Format 6 is similar to 5, but it does not treat every word as a keyword (not every word in the block is indexed). All keywords have to be entered by the user in a separate block within the entry. Only words contained within this block are indexed. This format is ideal when the user has already extracted information from the text and wants to include keywords which are not already part of the text. In Formats 5 and 6, all markers and keyword blocks are non-printing and hidden from view in normal display mode. This keeps the screen clear of clutter. Formats 7 and eight are similar to 5 and 6, except they use markers that are visible in normal display mode, and that print out as well. Text-Base supports various other combinations, as well as user defined formats for entries. No matter which format is chosen, *Text-Base* makes it very easy to export texts into the word processing module of *Nota Bene*, or another word processing system.

Although the process of using Nota Bene's *Text-Base* may sound complicated, note that the user spends very little time constructing a functional data-base from raw ASCII texts. By no means is this system the solution to every problem encountered in literary research; it is, however, an approach which simulates very closely the scholar's manual procedures when working with literary data.

Conclusion

Sophistication of text-based management systems will probably increase to a much higher level as research in Artificial Intelligence results in new 'intelligent' data-base and query systems. For the literary scholar, the most practical system is one that can point easily to the location of data about a desired topic. In current systems, this is accomplished through the use of an index and keywords. While keywords seem to work quite efficiently in locating data, do not forget that they also reduce the information value of a text to a minimum. Remember Todorov's understanding of the text as being the best description of itself. Another warning regarding the man-machine relationship: the types of tasks that the machine and program can perform on any data are limited and finite. If one becomes too dependent on the machine for literary scholarship, than one limits tasks only to solving those kinds of problems that are solvable by the system, while disregarding those features which point to the text's semantic exhaustiveness.

Notes

[1] *Notebook II* is a product of PRO/TEM Software, Inc., Walnut Creek, CA, reviewed by Puglia (1986).

[2] *AskSam* is a product of Seaside Software, Perry, Florida. Reviewed by Puglia (1986).

[3] *Nota Bene* is produced by Dragonfly Software, a Division of Equal Access Systems, Incorporated.

References

Barthes, R. (1967). *Elements of Semiology*. London: Cape.

Chatman, S. (1978). *Story and Discourse*. Ithaca: Cornell University Press.

Sohl, J. P. (1979). "SAS: for Literary Data Processing." *Fourth International Congress on Computers and Humanities*. Conference paper on SAS (Statistical Analysis System).

Eco, U. (1979). *A Theory of Semiotics*. Bloomington: Indiana University Press.

Iser, W. (1978). *The Act of Reading*. Baltimore: The Johns Hopkins University Press.

Puglia, V. (1986). "TBMS: Database Power Unleashed." *PC Magazine*, *5* (20), 211-230.