Automatic markup of SGML text databases

John Lehman
University of Alaska Fairbanks

Background

The project upon which this discussion is based is the conversion of a large, full text database from its original form to SGML. This turns out not to be a unique activity, because the vast majority of the full text databases which have been input, whether they are here at Academia Sinica, or have been input in Thailand, Korea or the United States are either not marked up at all or use some form of proprietary markup. The question which this paper addresses is what does it take to convert a full text database from a non-marked up non SGML form to SGML using one of the standard document type definitions such as TEI. What I'm reporting on here is the experience of trying to convert a fairly large system and what we have learned in doing it.

A follow up series of questions involve what can be done automatically? In other words for what can we write computer programs to do the conversion, what can be done using unskilled labor, and the most difficult of all what requires skilled labor, especially subject specialists?

Project Background

The database with which I have been working is called the Wenger Eskimo database. It is sponsored by the Wenger family in Switzerland.

This database consists of a texts of the first contacts between Europeans and various Eskimo peoples in the Far North, Greenland, Northern Canada, Alaska and Siberia. There are several hundred books which have been input in whole or in part into this database. The project was started more than five years ago by people in the archives section of our Library. These people had a good archival background but had never worked with a full text database project before, so the results were fairly typical of such first projects. They selected a propriety software system called Search Express to manage the data. The data entry standards were ad hoc at best. Each person involved did what he or she thought was appropriate in terms of how the data was entered, how it was structured and in some cases which sections of the text were selected. As a result even such basic things as paragraph markings are not standard throughout the original database.

Search Express, while it may have been a reasonable choice at the time, is produced by a company which now has very few customers. The software is obsolete, inflexible and generally does not do what one would expect a modern text management software package to do. It did include some structured markups for authors, titles, and figures and can indicate

special terms and so forth.

We decided about a year ago that we needed to convert the project into something which would be more standard. In June it was decided to convert it to SGML. At that time I became the technical director of the projects. SGML, as some of your know, is a very flexible system which required the use of a document type definition (DTD), which essentially defines what the structure of a document is.

The initial tendency of most people including the technical people working on our project was to go off and write there own specific DTD. However most industries have developed standard document type definitions, including the humanities and social sciences. The standard type document type definition for the humanities and for academic publications is that of the Text Encoding Initiative, or TEI.

There are a number of problems with TEI. The most obvious on which you encounter as soon as you attempt to read it, is that the full TEI description is about 1600 pages long in very small print. Much of this is not necessary for any individual corpus or individual document, because TEI was developed to be flexible to deal with any structure which is likely to be encountered.

As a result there have been several semi-official subsets of TEI developed. The two most popular are TEI light and what is know as bare bones TEI. Upon investigating these different document type definitions we concluded that TEI light which is used among other places for the Oxford text archives and the University of Michigan full text database projects would be sufficient for our project. Therefore we decided that we would convert all of our full-text data to TEI Lite so that in the long term it could be used by anyone with an SGML browser. Now those of you from Universities like Berkeley, Michigan, and Academia Sinica are used to projects that can have dozens or hundreds of people working on them simultaneously and budgets of many million dollars. The University of Alaska Fairbanks has about 6,000 full time students and the Wenger Eskimo database project is a large one for us-- we have 3 part-time staff (not counting myself who is quarter time with it). As a result we needed to be able to do this conversion as rapidly and inexpensively as possible without the staff esources that would exist at a larger institution.

Poverty can sometime be an advantage, since it forces one to be creative. It forced us to address the issue of whether we could convert the corpus automatically and if not what portion could be done automatically. The trial result, which is based on about fifty of these books which we are part way through converting, is that much of the initial conversion can be done by machine. As a matter of convenience, we are using a series of perl scripts rather than a proprietary conversion tool. The conversion from the original format was done in two steps. First we unloaded the database into a series of ASCII files, one for each book. We then applied perl scripts to the ASCII files which converts them to SGML as defined by TEI Light. This covers the basic conversion. At this point we can read these texts into

Author-Editor or any other SGML text editor and have valid SGML.

The perl scripts convert the structured markup which existed in the original, so author, title, dates, indexing terms, group names, figure definitions, anything which was defined in a standard form can be converted. That handles much of the basic conversion. We are also, optimistic that we will be able to do key word indexing automatically, although this is still in alpha testing.

There are however two major classes of markup that we have not found it possible to do automatically. Given our very low resources, we divided these into two groups based on who could to them. The one set of tasks were ones that programs couldn't convert because of the lack of structure in the original text databases. Essentially we needed to incorporate definitions of text structure, what were sections, what were chapters, sub-sections and so forth. In other words, we needed to be able to link the structure of the SGML document with the original paper text -issues of printers' layout.

The reason for separating these activities is that these can be given to clerical employees, work-study students or anybody else who is basically literate yet unskilled in the subject material. As a result it is relativelyinexpensive and fairly rapid to take care of this markup. and once this is done one has atext corpus which is usable in SGML form. In other words somebody can at this point use SGML browsers to read it or they can do searches by keyworkds or they can do searches for any topic indexing which was already in the original. They can thus get about 60 to 75 percent of the use of the text that would be poosible to get. In other words they now have the equivalent of the paper text with the ability to do some automatic searching.

What they do not have is much of the higher level mark up is needed to do topic indexing or term classification or hypertext links, which require an understanding of the content. These still need to be done by specialists.

So essentially we have a situation were the initial basic structural conversion can be done automatically, the structured markup, text structure, and links with the paper text can be one by unskilled workers leaving only the markup for real intellectual content such as keyword indexing, which requires specialists.

In other words, about 85% of the conversion work can be done without involving scholars. The next stpes in terms of manual markup, which we are probably not going to carry out on our project, can be divided into two classifications: those which need to be done entirely manually and those which can be done using some sort of computer aided support for the knowledge workers. An example of the computer aided work would be the project described at Academia Sinica where grammatical categories could be tentatively identified by means of an analysis program but then required a skilled human to check the output of the

program and identify which definitions have been make correctly -- in other words the program may be 80% correct and much of the tedious work can be done by machine, but you still need a skilled human to review the output. Based on some preliminary investigations it looks as if a relatively high proportion of markup for at least some forms of text could be aided by! this sort of decision support system.

As an example using Chinese Buddhist texts, one of the problems is the identification of sequences of characters used to translate or transcribe Sanskrit terms versus characters used in their original Chinese meaning. So, for example, if you encounter the character "Guan" as in "Guan-yin" does that mean "to look" or is that part of a translation of Avalokitesvara? By searching text with the help of a dictionary of Buddhist terms, tentative identification of the Sanskrit terms can be made by a computer, but it will still need be determined by a human editor whether the combination of characters is a technical term or happens to be two normal Chinese characters juxtiposed.

Summary

It appears then that much of the work in markup that requires scholarly input can be aided by computers even though it can not be completely automated. That will of course leave a certain proportion which will have to be done totally manually, and for that portion the difficulties of modifying the academic reward structure are probably the main impediment for producing the quality of text we would like.

Thus, the answer to the question as to how much created of marked-up full-text databases cost depends on how good we want our text. There appear to be a number of adequate ways in which we can significantly reduce the cost compared to what it would be if we did the markup totally by hand or even mostly by hand.