Hans Paijmans – Sander Wubben

Preparing Archaeological Reports for Intelligent Retrieval

Abstract: This paper describes the current state of the Open Boek information retrieval system for archaeological papers and reports in the Dutch language. The system focuses on the recognition of phrases that contain chronological and geographical references and measurements. In the course of its development, we have experimented with both Memory Based Learning and rule based techniques and we will describe the performance of both approaches for the recognition of chronological references.

Introduction

Elsewhere (Paijmans / Wubben 2007) we described the general principles and constraints that governed our approach to the problem of information retrieval in Dutch archaeological texts. Essentially we will not try to create or use a general ontology or other such "grand design" to interpret the contents of a text. Instead, we will try to solve the recognition of each semantic class on its own merits, strive for a satisfactory performance and then go on to the next class. If nothing else, we will create a robust system that is immediately useful for the archaeologist, as long as he/she does not expect one hundred percent perfection. However, as the performance of Information Retrieval in general never comes near the 100% mark, this in itself should not be an insurmountable obstacle. As a case, let us consider an institution such as the RACM[1], where a large number of papers and reports about archaeological excavations, site surveys and similar documents are stored digitally. Access to the information in the reports is by a collection of separate databases in which relevant attributes of the documents are either entered by human operators or occasionally by a rudimentary keyword index. Although there is traditionally much activity in the archaeological world in the field of typologies and controlled dictionaries, and hence there is an urgent need for so-called reference collections that support such typologies (Lange 2004), there is no agreement on how to apply this knowledge base to information retrieval and existing documents. Existing projects already do create a more involved XML markup for documents, based on CIDOC CRM, and so on, by J. Holmen and his collaborators (Holmen / Ore / Eide 2003), but there is no automatic extraction from instances in the text into the tags envisaged at this stage, while the current status of the project is unclear.

The needs of the archaeologist is concisely expressed as "What, when, where" and if we create a system that allows for these relatively simple questions, we will have the satisfaction of having addressed the primary needs of archaeological retrieval.

Our approach is to begin with the what; i.e. keyword access. In the technical sense this is not a major problem, as there are many good retrieval systems for plain text. Then we proceed to extract the semantic content needed for the when and where in stages. To do so, we are constructing an information retrieval system, *Open Boek*, that automatically extracts, translates and indexes such attributes from the text. At the moment, this data is used exclusively for retrieval, but in the long run the findings will be used for more involved operations, notably the identification of objects and collection of the data that is relevant for those objects. In this paper we present our progress in the recognition and interpretation of chronological data.

Open Boek: the Documents

The design of *Open Boek*[2] hinges on a few requirements. As we already mentioned, the system should be both immediately usable from the beginning of the project, while remaining open to additions and

[1] Rijksdienst voor Archeologie, Cultuurlandschap en Monumentenzorg, the central authority that collects data and coordinates archaeological activity in the Netherlands

[2] The programs and documentation can be found online at http://www.referentiecollectie.nl/Openboek

changes. Bearing this in mind, we adopted a tight modular approach, where the modules ("experts") communicate by ASCII files, using Unix text tools where possible. This may have had some impact on performance, but it certainly makes it easier to inspect intermediary results. Also, it invites experimenting with different modules that have similar functionality. Of course, our own programs, and the programs on which Open Boek depends, are all Open Source, and where possible licensed under the GPL.

The Original Documents

An important constraint is the format of the original documents. We had access to a large collection of thousands of reports in all kinds of formats, which we divided into three groups:

1. The largest portion of these, about two thousand reports of approximately 50 pages each, were originally typed on paper, and later scanned, OCR-red and stored as PDF. In these files, the "image" of every page was paired with an "invisible" ASCII text that could, however, be easily extracted and indexed. The problem arose of how to display the retrieved pages. The original PDF images contain all sorts of pictures, tables and drawings, but we did not address the technical problem of highlighting keywords or the addition of links in that PDF representation. Instead we converted the contents to HTML for that purpose. Consequently, the following problems arose:
   A. One alternative, the omission of the image of the page, and the display of only the ASCII text as HTML gave the opportunity to highlight and add links, but necessarily omitted most visual content such as images and formatting.
   B. The second option consisted of the projection of the HTML-ized ASCII over the image. This combines highlighting, links and visual content, but the result in the browser sometimes looks messy.
2. Another large portion of the files was already written using a word processor and stored as PDF. These files translated relatively easily to HTML, combining highlighting, links and images. However, the rendering was not always satisfactory.
3. A third group of documents consisted of hundreds of reports written by individual archaeological bureaus. These were stored on as many CDs and almost always produced by Microsoft software. Without a doubt every CD contains a highly artistic multimedia feast with sounds and movies, but it was absolutely impossible to extract the original reports without a time-consuming process of analysing the contents by hand, defeating the purpose of automated indexing and retrieval. Even if the "central" document could be identified, Microsoft's OLE framework essentially prevented extraction of the relevant data, at least with the tools that we used. So we limited ourselves to the PDF format and accepted for the moment the fact that the display sometimes was not as it should be. This, however, is a purely technical problem that can be solved at any time by buying appropriate, but expensive software.

Open Boek: Processing

As a first step, the PDFs are converted to individual HTML pages and separate images, keeping as much of the original layout as possible. Then, the text proper is extracted from the HTML. One typical database, scanned from paper and OCR-read, consisted of 750 PDF documents (1.7 GB) extracting to 50 MB text, contained in 30,000 pages and as many images. The HTML is extracted from the PDF files using pdftohtml[3], from which the final ASCII text is produced. For normal text this poses no particular problems, but the structure of a table becomes a casualty. Also, text in columns loses some of its coherency, which could be a problem when interpretation depends on text windows, as is usual in MBL (Memory Based Learning). The HTML tags and the text proper are stored in separate files, that are combined only when the page has to be rendered in a browser. This "standoff" notation makes it easy to add tags in a later stage, e.g. to mark chronological or geographical content.

Keyword Indexing

The documents are indexed on the document level and at the page level. The purpose of this twofold indexing is that combinations of keywords can be applied at both levels. We used the vener-

[3] http://pdftohtml.sourceforge.net

able SMART program, developed between 1965 and 1970 by Salton[4], which is still performing very well in the TREC contests (Buckley 2005). SMART is an implementation of the Vector Space Model (Salton / McGill 1983), which essentially retrieves documents on keyword combinations and, most importantly, ranks them according to some measure of relevance.

The SMART program offers several distinct weighting methods for individual words and we are still debating which is the best for this particular purpose. In any case, it serves as a fast and reliable indexing and retrieval engine and so can be the basis for a very usable document retrieval system. The creation of the keyword indexes for a database of this size is typically a matter of a few minutes on a modern PC running SuSE Linux.

Indexing of Numeric and Geographic Features

If SMART takes care of the *what*, the problems remain of the *when* and *where*. The indexing of geographical features is as yet in its experimental stage, although searching within a radius is already functioning. The purpose of this indexing is twofold. Firstly, to be able to search for such locations in terms as *"...within a circle of ten kilometres round Amersfoort..."* or *"...inside the county borders..."*. The second is disambiguation: *which county?* in the last example. A monument could, for instance, be called *"Loevenstein Castle"* and exist in a database somewhere with exact location, coordinates and so on. In the text of a document, however, it could be referred to as *"the castle"* or even *"the building"*, but our system should still be able to identify the castle from context and add precise information.

Chronological Indexing

The modules that handle chronological and geographical indexing are based on MBL. They create an index with all dates occurring in the document and store them as periods in the index-file. The location in the document is also tagged, although this is not essential. The final result is that the system "knows" that a particular year or period lies within the Middle Ages, or in the twelfth century, or the XIIth century, or whatever phrase is used in the document, and so is able to present texts relevant for that particular date.

The chronological indexing proceeds in three steps. First, the candidates for classification are collected by a numeric preparser. This preparser recognizes not only items like 2 and 100, but also the written, the Arabic and the roman cardinals and ordinals like the Dutch equivalents of two, second, 2nd or 2-nd, VI, VI-th, etc. In this phase, a list with names such as *"Middle Ages"*, *"iron age"*, *"Roman period"* is consulted, and the corresponding phrases are also flagged as chronological phrases.
The third and last phase is the normalization and the creation of the index proper. This includes assignment to BC or AD, and the decision whether the expression contains a single year, or is a period. "Between 1200 and 1300" obviously is a period, but so is "third century". More complicated are expressions such as "between the first century BC and the year 500", and we are still working to perfect our scripts to parse all possible combinations.

F-Score beta = 1 microav.	93
F-Score beta = 1 macroav.	87
AUC, microav.	96
AUC, macroav.	92
overall accuracy	94

Tab. 1. F-scores, area under the ROC-curve and overall accuracy obtained with TiMBL.

Memory Based Learning

For the MBL we used TiMBL 5.1 (Daelemans et al. 2004)[5], a decision-tree-based implementation of k-nearest neighbour classification (KNN). KNN classification is a method of classifying objects based on the closest training examples mapped in the feature space. TiMBL uses indexes in the instance memory extensively and therefore can handle discrete data and large numbers of various examples well (Daelemans et al. 2004).

First, we experimented with TiMBL, to obtain the optimum settings for this classification task. These settings were used to perform a tenfold crossvalidation test on the remaining data (22,563 instances).

[4] For more information on SMART and a tutorial see: Paijmans 1999.

[5] Available from http://ilk.uvt.nl

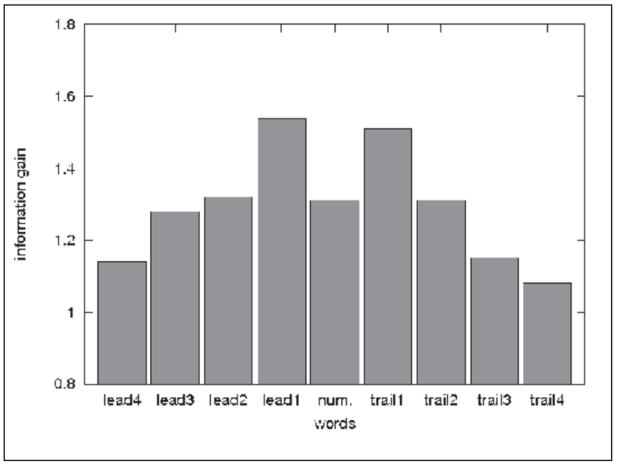

Fig. 1. Information gain of the nine features.

The numeric classes are based on CIDOC CRM. We are still debating whether these classes are the optimum solution for this scheme.

The performance of such systems is generally expressed in the recall (number of correct and positive predictions divided by the number of positive instances) and the precision (number of correct and positive predictions divided by the number of positive predictions). The F-score and the area under the ROC-curve (AUC) are different ways to express the precision and the recall in a single figure. *Tab. 1* shows these figures, using both micro- and macro-averaging. With an estimated total of 94% of the instances classified correctly, the MBL-component performs well and implementing it is therefore acceptable. Classes that do not perform very well are generally the smallest classes. As we are primarily interested in the large chronology classes, the performance in that field should even be better.

As demonstrated in earlier research (Buchholz / van den Bosch 2000), *Fig. 1* shows us that the closer the word is to the focus, the higher its information gain becomes. This means that words closer to the focus (in our case a numeric) are more important for the classification of that numeric. When at equal distance, words in front have a slightly higher information gain value than trailing words. Of particular interest is the fact that the information gain of the numeric itself is actually lower than the information gain of its direct neighbours. This means that the numeric itself contributes less to its classification than the words directly before or behind it. These results stress the ambiguity of numerics in archaeological texts and the need to use context to disambiguate the numerics before the numerical information contained within can be made explicit.

Retrieval

The keyword retrieval is based on the Vector Space Model, but offers three weighting methods: Boolean, frequency-based and atc. ("atc" weighting is a tf.idf weight. A tf.idf weight takes in account the frequency of a term within the document (tf) and the number of documents with that term (df). The atc variant also takes the length of a page into account). The queries are resolved by SMART itself.

More interesting is the processing of a chronological query. As we have observed, the various indexing modules except SMART create simple ASCII files, where the information about chronology and other classes is stored as plain text.

At retrieval, the user enters a simple expression that is either a single year (500), a period (500–1500), the name of a period ("Middle Ages" or even "200 BC – Middle Ages"), that are compared to the periods in the time index. He can also indicate whether his query should completely encompass the years and periods in the file, or that overlap at one or the other end is allowed. In the first case, the query "500–1500" will retrieve all pages on which references to years and periods within the Middle Ages (including the Middle Ages itself) occur. In the second case, also pages that refer to periods beginning before the Middle Ages, but ending within 500–1500, or conversely, periods that begin in the Middle Ages, but continue after 1500.

We will conclude this section with a small demonstration of the information that can be gleaned from the reports if and when the chronological information is made clear. From the current system, it is already relatively easy to map chronological references as in *Fig. 2*, where references to all years and periods within 1000 BC–2000 AD are shown in a database with reports on Dutch archaeological sites.

A few interesting features are visible from left to right. First, the plateau caused by frequent references to the "Iron age", defined as between 800 BC and 50 BC. Then a surge that starts sharply with the year 0 AD and rapidly falls off. This surge is caused by the incorrect classification of page numbers or paragraph numbers as years. The other spikes in the graph are caused by the human tendency to gravitate to "round" numbers. This is very visible in the years 50 BC, 500 AD and 1000 AD and to a lesser degree every 100 years. The Middle Ages in themselves are visible as another plateau in the

graph; caused by the frequent use of the term "Middle Ages". The activity then is lower for the next few hundred years, but rises a bit towards the year 2000 as a result of bibliographic references, that of course also include the year of publication.

Open Boek: Completion

We mentioned already that the recognition of numeric values and geographical references is

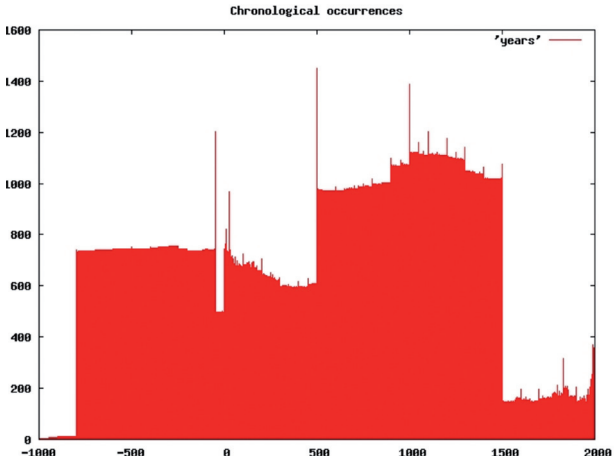

Fig. 2. References to years between 1000 BC and 2000 AD in RACM reports.

only the first step in the creation of the Open Boek system, and that the project has two very distinct goals. The first is to build a text retrieval system, that incorporates modern techniques for the interpretation of certain semantic classes, such as chronology or geography. To complete this stage, we have the following tasks before us:

- The current task, scheduled for completion in late spring 2007, is the subsystem that recognizes and disambiguates references to monuments, and adds the correct coordinates. This task is sponsored by KICH (KennisInfrastructuur CultuurHistorie) and the Nrc (Nationale Referentie collectie). We will use much the same approach as for the numeric data. In fact, the current system is already able to recognize spatial coordinates and display the corresponding area using Google Maps, or to retrieve pages that refer to locations within a certain distance of location X.
- Also, an Open Source stemmer for Dutch should be selected and implemented, to reduce the number of keywords.
- If and when performance becomes a problem, indexes and other data should be stored in a SQL-database, but that entails drastic redesigning of the system. Currently, most index files are plain ASCII, and are processed by the standard Unix text utilities.

Still more interesting (and more difficult) is the final task set to us: to carry the interpretation of NL (Natural Language) text to the point that we can identify phrases, passages and images that refer to (archaeological) objects mentioned in the text. We will describe our ideas and approach to that problem in a different paper.

Acknowledgements

This work was supported by now (Nederlandse organisatie voor Wetenschappelijk Onderzoek) and CATCH (Continuous Access To Cultural Heritage) under grant 640.002.401. No Microsoft software was used for the experiments mentioned in the paper or for the preparation of the paper itself.

References

Aha / Kibler / Albert 1990

D. W. Aha / D. Kibler / M. K. Albert, Instance-based Learning Algorithms. Machine Learning 7, 1990, 37–66.

Buchholz / van den Bosch 2000

S. Buchholz / A. van den Bosch, Integrating Seed Names and N-Grams for a Named Entity List and Classifier. In: Dybkjaer 2000, 1215–1221.

Buckley 2005

C. Buckley, Looking at Limits and Tradeoffs: Sabir Research at TREC 2005. In: Harman / Voorhees 2005.

Cost / Salzberg 1993

S. Cost / S. Salzberg, A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. Machine Learning 10, 1993, 57–78.

Daelemans et al. 2004

W. Daelemans / J. Zavrel / K. van der Sloot / A. van den Bosch, Timbl: Tilburg Memory Based Learner, version 5.1, reference guide. ilk technical report 04-02 (Tilburg 2004).

Dybkjaer 2000

L. Dybkjaer (ed.), Proceedings of the Second International Conference on Language Resources and Evaluation (Athens 2000).

Harman / Voorhees 2005
D. Harman / E. Voorhees (eds.), The Fourteenth Text Retrieval Conference (Washington 2005).

Holmen / Ore / Eide 2003
J. Holmen / C.-H. Ore / Ø. Eide, Documenting Two Histories at Once: Digging into Archaeology. In: Computer Applications and Quantitative Methods in Archaeology 2003. BAR International Series 1127 (Oxford 2003) 221–224.

Lange 2004
A. G. Lange, Reference Collections, Foundation for Future Archaeology (Amersfoort 2004).

Paijmans 1999
J. J. Paijmans, Explorations in the Document Vector Model of Information Retrieval. PhD. Thesis (Brabant 1999).

Paijmans / Wubben 2007
J. J. Paijmans / S. Wubben, Memory Based Learning and the Interpretation of Numbers in Archaeological Reports. In: M.-F. Moens / T. Tuytelaars / A. P. de Vries (eds.), Proceedings of the 7th Dutch-Belgian Information Retrieval Workshop (Leuven 2007) 51–56.

Salton / McGill 1983
G. Salton / M. J. McGill, Introduction to Modern Information Retrieval (New York 1983).

Hans Paijmans

Institute for Knowledge and Agent Technology
Tilburg University
Warandelaan 2
5037 AB Tilburg, The Netherlands
paai@uvt.nl

Sander Wubben

RACM Amersfoort
Kerkstraat 1
3811 CV Amersfoort, The Netherlands