Preparing archeological reports for intelligent retrieval

Hans Paijmans[1] – Sander Wubben[2]

[1]Tilburg University , Tilburg, the Netherlands
[2]RACM, Amersfoort, the Netherlands

Abstract: This paper describes the current state of the Open Boek information retrieval system for archeological 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.

Keywords: Memory Based Learning, information retrieval, chronology

1 Introduction

Elsewhere (cite anonymized) we described the general principles and constraints that governed our approach to the problem of information retrieval in Dutch archeological 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, but we 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 archeologist, as long as he/she does not expect a hundred percent perfection. But as the performance of Information Retrieval in general never comes near the 100% mark, this is 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 archeological 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 entered by human operators or, sometimes, by a rudimentary keyword index. Although there is traditionally much activity in the archeological world in the field of typologies and controlled dictionaries, and although 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. This is typical for the archeological scene; there exist projects to create a more involved XML markup for documents, based on CIDOC/CRM, e.g, by J. Holmen and his collaborators, (Holmen et al., 2003), but here no automatic extraction from instances in the text into the tags is envisaged, and the current status of the project is not clear.
The needs of the archeologist 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 archeological 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. For this, we are constructing an information retrieval system, **Open Boek**, that automatically extracts, translates and indexes such attributes from the text. At the moment, such data are 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 are relevant for those objects. In this paper we present our progress in the recognition and interpretation of chronological data.

2 Open Boek: the documents

The design of **Open Boek**[2] hinges on a few requirements. As we already mentioned, the system should be both

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

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

immediately usable from the beginning of the project, and remain open for additions and changes. Therefore 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.

2.1 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 divide in three groups:

1. By far the largest portion of these (about two thousand reports of approx. fifty pages each) were originally typed on paper, and later scanned, OCR-red and stored as PDF. In such files, the "image" of every page was paired by an "invisible" ASCII text that however could be easily extracted and indexed. The problem here was the display of the retrieved pages. The original pdf-images of course 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. However: this gave rise to the following problems:
A. One alternative, the omission of the image of the page, and the display of only the ASCII text as HTML gave the opportunity of highlighting and links, but omitted necessarily most visual content such as images and most 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 wordprocessor and stored as PDF. Such files translated relatively easy in HTML, combining highlighting, links and images. Still, the rendering is not always satisfactory.

3. A third group of documents consisted of hundreds of reports written by individual archeological 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, movies and everything, but it was absolutely impossible to extract the original reports without a timeconsuming process of analysing the contents by hand, defeating the purpose of automated indexing and retrieval. But even if the "central" document could be identified, Microsofts 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.

3 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 was extracted from the HTML. One typical database, scanned from paper and OCR-red, consisted of 750 pdf documents (1.7 GB) extracting to 50 MB text, contained in 30.000 pages and as many images. The extraction of the HTML from the PDF files is done by 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 "stand-off" notation makes it easy to add tags in a later stage, e.g. to mark chronological or geographical content.

3.1 Keyword indexing

3 http://pdftohtml.sourceforge.net

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 venerable SMART program, developed between 1965 and 1970 by Salton[4], that is however still doing very well in the TREC contests (Buckley, 2005). SMART is an implementation of the Vector Space Model (Salton and 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.

3.2 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 withhin a radius is already functioning. The purpose of this indexing is twofold: first to be able to search for such locations in terms as *"...within a circle of ten kilometers round Amersfoort..."* or *"...inside the county borders..."*. The second is disambiguation: *which county?* in the last example. A monument could, e.g. 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.

3.3 Chronological indexing

Our modules to do the 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 (see table 3). 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 in the XII-th 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 etcetera). In this phase, also a list with names *("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 as "between the first century BC and the year 500", and we are still working to perfect our scripts to parse all possible combinations.

3.3.1 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).

---

4 For more information on SMART and a tutorial see: (Paijmans, 1999)

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

F-Score beta = 1 microav.	0.93
F-Score beta = 1 macroav.	0.87
AUC, microav.	0.96
AUC, macroav.	0.92
overall accuracy	0.94

Table 1: F-Scores, Area under the ROC-Curve and overall accuracy obtained with TiMBL

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). The numeric classes are based on CIDOC/CRM; we are still debating whether these classes are the optimum solution for this scheme.
With a total of 94% of the instances classified correctly, the MBL-component performs well and implementing it is therefore acceptable. Classes that don't perform very well are, not surprisingly, generally the smallest classes; as we are primarily interested in the large chronology class, the performance in the field should even be better.

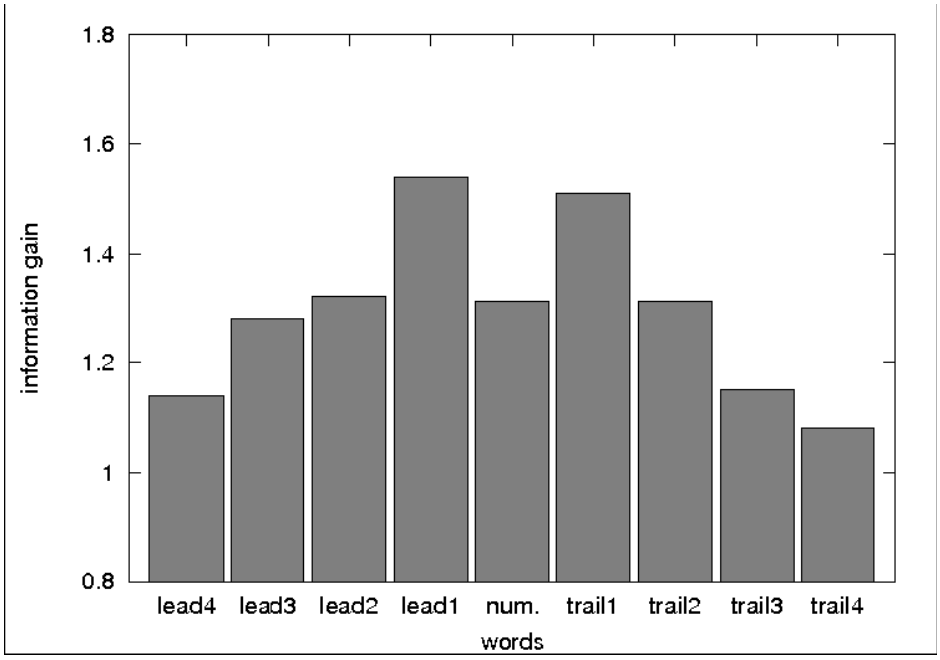

Figure 1: Information gain of the nine features

As demonstrated in earlier research (Buchholz and van den Bosch, 2000), figure 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. Something that makes our situation interesting, 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 in front or in the back of it. These results stress the ambiguity of numerics in archeological texts and the need to use context to disambiguate the numerics before the numerical information therein can be made explicit.

3.3.2 Retrieval

The keyword retrieval is based on the Vector Space Model, but offers three weighting methods for the results

of the search: boolean, frequency-based and atc-weighted ("atc" weighting is a tf.idf weight that takes in account the length of a page).
The queries are resolved by SMART itself.

More interesting is the processing of a chronological query. As we have seen, the various indexing modules (except SMART) create simple ASCII files where the information about chronology and other classes is stored explicitly (see table 2). In the example we see the file, the page number, the beginning of a time period and the end of that period.

Document-page	start period	end period
Aardenburg-1	+19900101	+19901231
Aardenburg-3	+19400101	+19451231
Aardenburg-3	+19450101	+19451231
Aardenburg-4	-50101	+500231

Table 2: Some lines from the time index

At retrieval time 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 "200BC - 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.

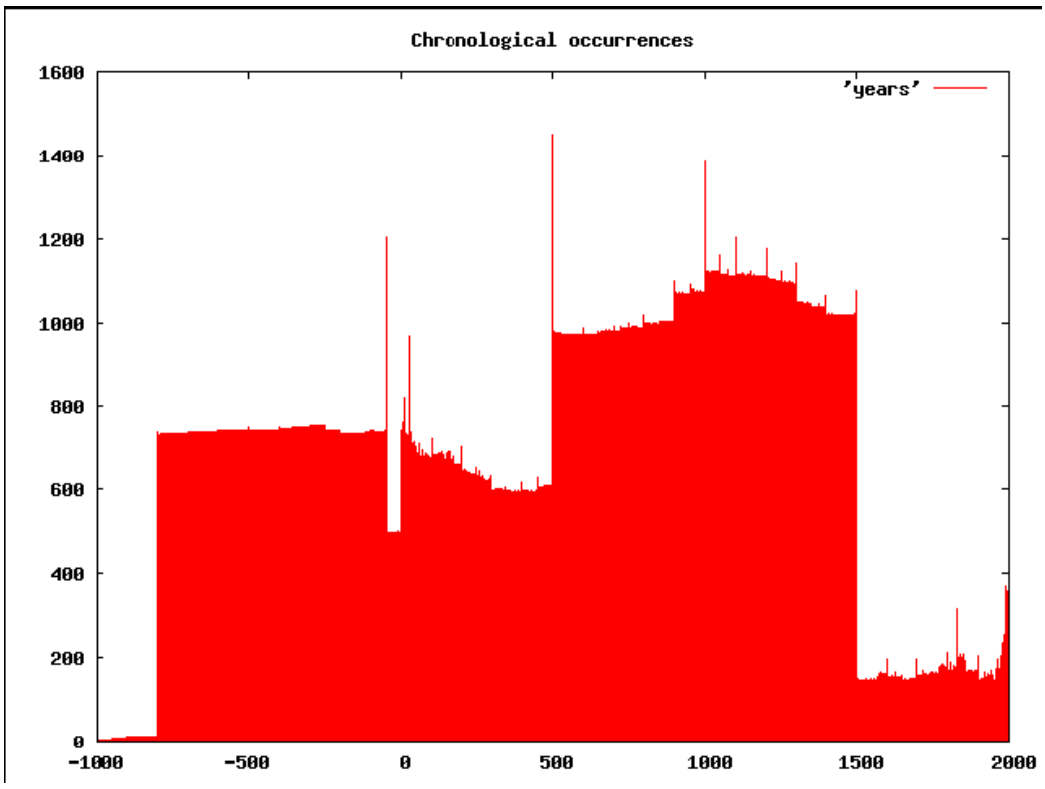

Figure 2: References to years between 1000 BC and 2000 AD in RACM reports

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 explicit. From the current system it is already relatively easy to map chronological references as in fig. 2, where references to the all years and periods within 1000 BC-2000 AD are shown in a database with reports on Dutch archeological 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 e.g. 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.

4 Open Boek: completion

We mentioned already that the recognition of numeric values and geographical references are only the first steps 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 add 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 Googlemaps, 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 before us: to carry the interpretation of NL (Natural Language) text to the point that we can identify phrases, passages and images that refer to (archeological) 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

David W. Aha, Dennis Kibler, and Marc K. Albert. 1990. Instance-based learning algorithms. Machine Learning, 7:37–66
S. Buchholz and A. van den Bosch. 2000. Integrating seed names and n-grams for a named entity list and classifier. In Dybkjaer , pages 1215-1221.
C. Buckley. 2005. Looking at limits and tradeoffs: Sabir research at TREC 2005. In Harman and Voorhees
Scott Cost and Steven Salzberg. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. Mach. Learn., 10(1):57-78.
Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. Timbl: Tilburg memory based learner, version 5.1, reference guide. ilk technical report 04-02. Technical report, Tilburg University.
L. Dybkjaer, editor. 2000. Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000).
D. Harman and E. Voorhees, editors. 2005. The Fourteenth Text REtrieval Conference (TREC-14). National Institute of Standards and Technology.
J. Holmen, C-H. Ore, and O. Eide. 2003. Documenting two histories at once: digging into archaeology. In CAA 2003 - Computer Applications and Quantitative Methods in Archaeology. Bar International Series 1127 2004.
A.G. Lange, editor. 2004. Reference collections, foundation for future archaeology. Rijksdienst voor Oudheidkundig

bodemonderzoek, Amersfoort..
J. J. Paijmans. 1999. Explorations in the Document Vector Model of Information Retrieval. Ph.D. thesis, Katholieke Universiteit Brabant.
G. Salton and M. J. McGill. 1983. Introduction to Modern Information Retrieval. McGraw-Hill New York [etc.] -448 pp.