Putting the Dutch PAROLE Corpus to Work

P.H.J. van der Kamp and J.G. Kruyt
Institute for Dutch Lexicology INL
P.O. Box 9515, NL-2300 RA, Leiden, The Netherlands
{kamp, kruyt}@inl.nl

Abstract

We discuss the activities towards the development of the retrieval application of the Dutch PAROLE Corpus. Compared to the other corpora developed by INL, the PAROLE Corpus has been encoded with more extended types of metadata, conformant to the TEI standard for text encoding. A search engine and a web-based user interface, both newly developed by INL, provide the user with the functionality to explore the corpus, not only at the level of the text, but also at the level of the metadata or a combination of the two. In view of our experience with corpus retrieval, we did not follow the complete system development cycle, but used an alternative method instead.

1. Introduction

Between 1993-1996, the Institute for Dutch Lexicology (INL) developed three text corpora, which can be consulted over the Internet by use of a retrieval system (www.inl.nl/eng/corp/corp.htm). Hundreds of subscribed users consult the corpora for various purposes: for lexicography and lexicon building, for a variety of research purposes in the fields of linguistics and social studies, and for university courses in corpus linguistics (Kruyt, 1998).

We have nearly finished the Internet application of the Dutch PAROLE Corpus, which was built in the framework of the EC-funded LE-PAROLE project (1996-1998); its characteristics are reported on in section 2. Starting from user needs, the functionalities of a retrieval system were defined (section 3). We then decided to build our own retrieval system rather than using an available one, for several reasons explained in section 4. The technical choices and the user interface are described in the sections 5 and 6. The paper concludes with a broader perspective for this application.

2. The Dutch Parole Corpus

The Dutch PAROLE corpus is a collection of present-day Dutch texts, containing around 20 million words. The texts were obtained from various publishing houses and other third parties, which implied that their use was to be contractually defined (copyright). Use is permitted for non-commercial research purposes only, and access is restricted to rather small texts fragments, with proper reference of the source.

In order to give flexible access to the data, the texts are encoded with several types of metadata. The texts are classified according to the PAROLE topic domain categories (leisure, history, etc.) and to the PAROLE medium categories (newspaper, book, periodical, etc.). For each text, the text structure and typography are encoded up to level 1 of the Corpus Encoding Standard. The words of the texts are encoded with their PAROLE Part of Speech (PoS) (word class) and with their headword (dictionary entry form). The format of the encoding is conformant to the TEI standard for text encoding (an SGML application). This encoded corpus is the input for the retrieval system (cf. sections 5 and 6).

It required a substantial amount of work to give these characteristics to the corpus. First, we had to convert the texts to a format that was suitable for further processing. Most texts were delivered as a WordPerfect, Word or plain text file and therefore converted to plain text files. The typographical encoding and/or text structure encoding were replaced by the so-called INL format: a mixture of SGML/HTML-like tags, meant to preserve the original information in a software-independent way. In some cases additional processing was performed in order to extract the data we needed.

Secondly, the intermediate INL format was replaced by mark-up according to the TEI standard. Based on typographical features and the available encoding, a formal description was made of how the appropriate TEI tags for text structure and typography had to be included in the text. For each description conversion software was made, using an advanced 'search and replace language'-compiler developed by the INL. This software not only adds the TEI tags but also merges the text with the accompanying handmade or semi-automatically generated header file. Static information, such as medium and topic, was already present in the header. Dynamic information such as the number of words, the tags used in the text and their frequency was inserted in the header by shell and Perl scripts. The last stage in this process was validating the TEI encoded text for which we used James Clark's SGML parser nsgmls.

Thirdly, after automatic sentence tagging, the words of the texts were automatically encoded with their PoS and headword, by use of a PoS-tagger and a PAROLEX-lexicon of ca. 240,000 entries (Does & Voort van der Kleij, 2002). The lexicon is our former coarse-grained DutchTale-lexicon (Voort van der Kleij & Kruyt, 1997), which has been converted to the fine-grained PAROLE tagset and extended with lexical entries from the Dutch PAROLE-lexicon. The tagger is a combination of statistically-based taggers, which makes use of a training corpus of 100,000 words. Much effort was spent on improving the quality of the training corpus, on the optimization of the tagger and on improving the quality of the tagged and lemmatized Dutch PAROLE Corpus by checking certain selected tag assignments manually and improving other tag assignments by a rule-based correction process. After this process, the texts were validated again.

3. Functionalities: User Needs

Generally speaking, our users (cf. section 1) need a retrieval system that gives flexible access to linguistic information in user-defined collections of data. A number of functionalities already available in our former internet-corpora need to be maintained (all searches are recorded, so we can trace user needs). With respect to search facilities, the major ones are: searches for single words and multi word units by use of Boolean operators and proximity options; searches at the levels of not only character string, but also of PoS and headword with its paradigm (possible due to the PoS and headword encoding; cf. section 2); queries with combinations of those levels; searches for predefined word classes (e.g. present and past participle) and phrasal categories (e.g. NP, PP) that can be customized by the user; the possibility to restrict a search to a user-defined subcorpus by use of the metadata 'topic', 'medium' (cf. section 2) and date, or by a selection of individual texts. With respect to the output: intermediate screens with frequency lists of word forms and/or headwords, with the possibility of selecting items; screens with concordances which can be extended to a longer quotation with proper source reference, flexible sorting facilities, etc. (cf. Kruyt, 1998). Essentially new to the Dutch PAROLE corpus is the encoding of text structure and typography, which enables the user not only to search for these elements, but also to use them for defining a search domain within which the desired information is to be searched. Other new functionalities include concordances and quotations either with or without tags shown; data on distribution of word forms, headwords, PoS categories and text-structural tags calculated with respect to medium, topic, date and individual text sources; statistical collocations by use of different methods; the facility to filter undesired list items or concordances with adaptation of frequency data; easy navigation through list and concordance screens by the facility to jump to numbered lines; user-defined default and non-default settings; more possibilities to compile a user-defined subcorpus, saving of subcorpus definitions; extensive help functions and corpus documentation, also for potential users who have not subscribed yet. A more sophisticated functionality (called “patterns”) now enables the user to retrieve certain word classes, word groups, syntactic phrases and sentence types by means of 26 predefined complex queries, which can be customized by the user to create new ones. The predefined queries consist of sophisticated combinations of word form, headword, PoS and their features, TEI-encoding, Boolean operators, proximity, regular expressions, recursion, etc. As they are part of the advanced search facility, it was necessary to extend the query language which supports the whole functionality mentioned above. There was also a need for a more user-friendly interface (cf. section 6).

4. Why a *New* Retrieval System?

When the functionalities had been defined, we had to decide whether we would build a new retrieval system or use an existing one. Given the broad experience of the INL with corpus retrieval systems, we first considered the use of one of our former systems. We decided against it. These systems were built for corpora with no text-structure encoding and a coarse-grained PoS encoding. They were furthermore intended to be used over a Telnet connection, whereas the new system had to be web-based, thus requiring a client-server architecture. Porting the approximately 25,000 lines of (OpenVMS) Pascal code of the former applications to (Unix) C and simultaneously changing the architecture would have been an enormous effort.

Then, we considered the use of similar retrieval systems built by other institutions. Amongst the systems we investigated in the first quarter of 2001 were well-known systems as COSMAS I (http://www.ids-mannheim.de), IMS Corpus Workbench (http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench), BNC/SARA (http://info.ox.ac.uk/bnc) and Qwick (currently only available for members of the TRACTOR user community). The IMS Corpus Workbench is used in, for instance, the CETEMPúblico corpus project (http://acdc.linguateca.pt/cetempublico/whatisCETEMP.html). Two questions were important: do these systems meet our requirements and if not, can they be easily adapted to our needs. Information available on the Internet and consultation by email demonstrated that none of the systems provide all functionality desired for the Parole Internet Corpus. For instance, the IMS system produces only concordance lines, whereas we also need word or lemma lists from which further selections can be made. IMS and COSMAS I allow the user to select a certain corpus but the user cannot restrict the search domain to individual texts within that corpus. Qwick offers limited sort options where we need more advanced options. Given these limitations, it would be necessary to adapt the server and client software. Except for BNC/SARA, where the client software is available for registered users, none of the other institutions make the source code of their software available to others. In one case, Perl modules are available to interact with the software but these were rather undocumented. Our conclusion was that we had to build the software ourselves.

5. Technical Choices

When we started the development, we had to choose an appropriate technology to realize the client-side user needs (cf. section 3). For a web-based client a Java applet or JavaScript implementation is the most suitable technique. JavaScript is a (ECMA) standardized scripting language; Java is currently not an official standard but a de facto standard. Both languages are supported by web browsers although the level of support can vary. Despite the ‘write once, run everywhere’ motto of Java, we regularly get into difficulties running Java applications or applets from the web, because the proper version of the Java run-time environment (JRE) or Java development kit (JDK) had not been installed or included with the browser. From the user's perspective one of the requirements is that the effort to get the application or applet to work should be as small as possible; for example, just by upgrading his webbrowser the user should obtain the proper JRE. We were not sure whether this requirement could be achieved with a Java implementation, although we considered Java capable of realizing the functionality of the client. Before making a final decision, we investigated the feasibility of a JavaScript solution by implementing some of the more complicated functions. This small study showed that it

was possible to use JavaScript, but it also became clear that it would not be easy to develop scripts that would work properly with both our preferred browsers Netscape and Internet Explorer (IE). In spite of this, we decided to use JavaScript to implement the client. The Java licence problems between Sun Microsystems and Microsoft contributed to this decision.
After a few months of client-side development it became clear that the intended cross-platform compatibility could only be achieved at very high costs. For this reason we decided to develop for IE only: some functionalities were easier to implement and it is the most widely used browser.
The functionalities also lead to decisions that are more hidden from the user. Normal browser behaviour is that parts of a document are being displayed while document transfer is still in progress. Functionalities such as filtering, selecting (a range of) concordance lines or jumping to a specific line require that no data are displayed until all data have been received. For example, the user cannot jump to line 500 when the browser is receiving the first 200 lines of data. To achieve this, we decided to put the data in a HTML <table>-tag.
With regard to server-side development, we had to choose between the (Unix) file system or a (relational) database to store the indexes. We decided to use the former to avoid additional processing and also because the use of databases for corpus exploration purposes is not widespread.

6. User Interface

Section 3 described which functionalities the system has to provide to the user; this section is concerned with the question how the user can make use of these functionalities through the user interface. The text-based user interface of our former Internet corpora needed to be rather simple due to the equipment restrictions of our users at the time. This interface does no longer meet the current requirements of user-friendliness, which is why we developed a graphical user interface. As we do not want to bother the user too much with technical or installation issues (cf. section 5), we decided to develop the interface for software that is already present on most PC's: the web browser. In a web environment developers can be tempted to activate functionality by using all kinds of visual effects, simply because it is easy to implement. However, a user benefits from a transparent and consistent design, where visual effects contribute to an effective use of the interface, i.e. "make explicit and immediate many of the powerful options of the system" (Hearst, 1999). For example, due to the limited capabilities of the former text-based user interface, the user had to know that he could use Boolean operators and proximity options in a query. In our web interface, buttons make this functionality explicit.
In order to make the interface as transparent as possible, a leading principle was that the concepts used in the interface should be familiar to the user. For this reason we used the tab concept, which is common practice in software and web pages; the user can easily switch between searching, (search) results, subcorpus selection, default settings and help. The interface has a query screen for simple searching and one for complex searching, similar to web-search engines; the need for this distinction was based on an analysis of our users' queries.
After he has logged in, the user can choose between three options: simple searching (i.e. search for one type of information only, e.g. lemma or word form, etc.), complex searching (i.e. many more, and more advanced search options, including regular expressions, proximity, saving queries, patterns, etc.) and collocations. Via the tab 'subcorpus selection', which can be activated at several places in the interface (just like the other tabs), the query can be restricted to parts of the corpus. Results of all types of queries and subcorpus selections are presented in result screens. After intermediate result screens with several options for sorting and selecting items, the final result consists of concordances which can be extended to larger quotations, both visualized with or without tags. See section 3 for further details of the functionalities.
For the design of the interface we did not use traditional system-development methodologies (e.g. SDM, DSDM), which require many reports before a line of code can be written. It is hardly possible to describe the functionalities and the 'look and feel' of the user interface in a way which can easily be understood by users and developers. By use of painting software, we designed interface screens that came very close to the web version. The resulting pictures were thoroughly discussed with linguists before they were implemented as web pages. Depending on the functionality, alternatives were offered to the linguists from which they could choose the most convenient one. This way, we tried to cope with the problem of misunderstandings between technical and linguistic staff, with the problem that users have little knowledge of the technical options, and with the problem that users often do not know what they want exactly. Working this way, the interface became more and more transparent, consistent, and easy to use. Regular demo sessions with the real web pages showed that the design decisions were sound; only minor modifications were necessary.

7. Future Work

This paper described a flexible retrieval system for the Dutch PAROLE corpus, which will be fully operational in the summer of 2004. Our users will be requested to evaluate the system, as it also has relevance to a new INL project, the "*Integrated Language Database of 8th – 21st-Century Dutch (ILD)*". This project aims at creating a database covering the oldest up to the most recent Dutch language, which functions as a flexible instrument for a wide range of research into the Dutch language (and culture) throughout the centuries. In the database, corpus data, dictionary data and lexicon data will be linked and integrated. The user will have access to these types of data through one interface. The project is in an advanced conceptual phase (cf. Kruyt, 2000; Dalen-Oskam & Geirnaert & Kruyt, 2002; Depuydt & Dutilh-Ruitenberg, 2002); a prototype is now developed (Kruyt, 2004). The retrieval system for the Dutch PAROLE corpus serves as a model for access to the corpus component of the ILD.

Acknowledgements

The following colleagues substantially contributed to the development of the Dutch Parole Corpus and its retrieval

system (colleagues who are no longer in the department are marked with an asterisk).
EDP Department: Sonja Deutekom, Bart Hoogeveen, Wimjan Jansen van de Laak*, Pieter Masereeuw*, Nikos Massios, Dennis Schenk and Rob van Strien.
Language Database Department: Marjolijn van Bennekom, Katrien Depuydt, Jesse de Does, Hedy Ede van der Pals*, Arjen Figee*, Stephan Raaijmakers*, Tilly Dutilh-Ruitenberg, Wil de Ruyter, Fiona Thomson*, Marianne Veltkamp*, Boukje Verheij and John van der Voort van der Kleij.

References

Dalen-Oskam, K. van, D. Geirnaert & T. Kruyt (2002). Text Typology and Selection Criteria for a balanced Corpus: The Integrated Language Database of 8th–21st-century Dutch. In: Anna Braasch & Claus Povlsen, *Proceedings of the Tenth Euralex International Congress, Euralex 2002*, p. 401-406. See also http://www.inl.nl/eng/pub/euralex_dalen_eng.htm

Depuydt, K. & T. Dutilh-Ruitenberg (2002). TEI-encoding for the Integrated Language Database of 8th–21st-Century Dutch. In: Anna Braasch & Claus Povlsen, *Proceedings of the Tenth Euralex International Congress, Euralex 2002*, p. 683-688. See also http://www.inl.nl/eng/pub/tei.htm

Does, J. de & J. van der Voort van der Kleij (2002). Tagging the Dutch PAROLE Corpus. In: Mariët Theune, Anton Nijholt & Hendri Hondorp, *Computational Linguistics in the Netherlands 2001, Selected Papers from the Twelfth CLIN Meeting*, p. 62-76. See also http://www.inl.nl/eng/pub/clinproceedings.pdf

Hearst, M.A. (1999). User Interfaces and Visualization. In: Ricardo Baeza-Yates & Berthier Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, New York, p. 257-323.

Kruyt, J.G. (1998). Dutch Written Language Resources, their Users and Uses. In: *Proceedings First International Conference on Language Resources & Evaluation*, pp. 959-963. See also http://www.inl.nl/eng/pub/grancon.htm

Kruyt, J.G. (2000). Towards the Integrated Language Database of 8th-21st Century Dutch. In: *Revue française de linguistique appliquée* V-2 (Décembre 2000), 33-44. See also http://www.inl.nl/eng/pub/ildkruyt.htm

Kruyt, J.G. (2004). The Integrated Language Database of 8th-21st Century Dutch. In Proceedings LREC 2004.

Voort van der Kleij, J. van der & T. Kruyt (1997). Lexicon for a linguistic annotation of Dutch text, *TELRI Newsletter 5*, 32-35.