CHAPTER 19

WhiteLab 2.0: A Web Interface for Corpus Exploitation

Matje van de Camp[c,a], Martin Reynaert[b,a] and Nelleke Oostdijk[a]

[a]CLS / Radboud University Nijmegen, [b]TiCC / Tilburg University,
[c]De Taalmonsters, The Netherlands

ABSTRACT

The OpenSoNaR-CGN project set out to develop WhiteLab 2.0 for the online exploitation of the SoNaR-500 and CGN corpora. Important changes in comparison to the first version of WhiteLab are the addition of audio support and support for multiple corpora. The web interface has been redeveloped and adapted to accommodate these changes. At the backend, WhiteLab 2.0 comes with a new data importer and plugin for Neo4j, while also remaining compatible with BlackLab. Although performance of the new backend is not yet up to par with BlackLab, the investment in new technology that will likely be further developed is expected to make the application more future-proof and a great addition to the set of tools available to the humanities.

19.1 Introduction

Since the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN (Oostdijk 2000)) project set out in 1998 to compile a corpus of standard Dutch, the landscape of Dutch language resources has changed dramatically. At the turn of the century Strik et al. (2002) reported in a survey they conducted of Dutch language resources that they found the Human Language Technologies (HLT) infrastructure to be "scattered, incomplete, and not sufficiently accessible". Thanks to substantial investments by the Dutch and Flemish governments and research foundations in the STEVIN programme[1] (D'Halleweyn et al. 2006; Spyns and Odijk 2013) and the CLARIN-NL project (Odijk 2010) most of what are generally considered to be basic language resources are now in place and can be accessed in a common infrastructure.

[1] STEVIN was a five-year (2004-2009) joint Dutch-Flemish programme for Language and Speech Technology.

How to cite this book chapter:
van de Camp, M, Reynaert, M and Oostdijk, N. 2017. WhiteLab 2.0: A Web Interface for Corpus Exploitation. In: Odijk, J and van Hessen, A. (eds.) *CLARIN in the Low Countries*, Pp. 231–243. London: Ubiquity Press. DOI: https://doi.org/10.5334/bbi.19.

Since the focus of the STEVIN programme was on settling the pressing needs as they existed in the HLT community, its orientation was first and foremost towards users that had the necessary skills to handle the tools and the data. The CLARIN project, however, aimed to develop an interoperable research infrastructure for humanities researchers that work with language data and tools. The infrastructure should make it possible for them to find and access data and tools relevant for their research. Importantly, researchers should be able to apply available tools to their data in such a way that no technical background is needed or ad-hoc adaptations to the tools or the data are necessary.[2]

As the opportunity arose within CLARIN-NL to address the need for a corpus exploitation tool that would make it possible for users to access the large (500+ million-word) reference corpus of written standard Dutch (SoNaR-500 for short; Oostdijk et al., 2013), the OpenSoNaR project (Reynaert et al., 2014) was initiated[3]. It took its lead from other projects concerned with large national corpora, which successfully employed the latest online web-based technology, and developed WhiteLab as a frontend to the then new corpus indexer BlackLab[4] which had been developed by the former Institute for Dutch Lexicology (INL), now Institute for the Dutch Language (INT). Then, in 2015, as WhiteLab had proved its usability and user-friendliness through OpenSoNaR, with additional funding from CLARIN-NL through the OpenSoNaR-CGN project it was extended to add support for spoken language corpora. The resulting system, WhiteLab 2.0, makes it possible for users to access and exploit both SoNaR and CGN, either independently of each other or in combination. The combined corpora are now online under the new name OpenSoNaR+[5].

In this chapter we describe WhiteLab 2.0 and the interface to SoNaR and CGN. The structure of the chapter is as follows: in the next section, we introduce the two corpora in some more detail. In Section 19.3 we describe WhiteLab 2.0's architecture and provide a preliminary comparison of its newly developed backend with the existing BlackLab. Then, in Section 19.4 we turn to the user-functionality that it offers by describing the OpenSoNaR+ interface. In Section 19.5 attention is given to the performance and availability. Section 19.6 concludes this chapter.

19.2 The Corpora

The Spoken Dutch Corpus (Oostdijk, 2000) is a corpus of some 800 hours of speech, comprising a large number of samples recorded from adult speakers in the Netherlands and Flanders speaking standard Dutch. All data have been orthographically transcribed, annotated for parts-of-speech, and lemmatised. For a subset of the data phonetic transcriptions and syntactic annotations are also available. The metadata provide information about the speakers (e.g. age, sex, place of birth, educational background) and the recordings (e.g. duration, recording conditions, number of speakers). In order to allow for less technically savvy researchers to use the corpus without having to call upon the assistance of someone with programming skills, the COREX (CORpus EXploitation) software was developed (Oostdijk and Broeder 2003). It enables users to browse and search the corpus, and to view and export the results. Exploitation in COREX is limited to the transcriptions and annotations that are available for the full corpus. For the other annotation layers users are expected to make use of dedicated software, such as Praat[6] for phonetic transcriptions or Dact[7] for the

[2] http://www.clarin.nl/

[3] It was apparent that the dedicated software developed for exploitation of the Spoken Dutch Corpus would not be able to handle the amounts of data found in the SoNaR corpus.

[4] https://github.com/INL/BlackLab

[5] http://opensonar-cgn.science.ru.nl

[6] http://www.fon.hum.uva.nl/praat/

[7] http://rug-compling.github.io/dact/

exploitation of the syntactic annotations. Since all transcriptions and annotations are directly or indirectly aligned with the audio, the user can access the recordings from any point in the corpus. Searches can be conducted involving information from different annotation layers. The metadata may be used to further restrict a search to a specific subset. Results are presented in the form of concordance lines or, in the case multiple where content searches are executed on different subcorpora, frequency lists.

SoNaR-500 (Oostdijk et al., 2013) is a 540-million-word reference corpus of contemporary written Dutch. It includes a balanced collection of full texts representing a broad range of genres and text types, such as books (fiction and non-fiction), newspaper articles, and brochures, but also from the new and social media, such as discussion fora, chats, and tweets. The texts are original Dutch texts from the Dutch-speaking language area in the Netherlands and Flanders, or Dutch translations published in and targeted at this area. All texts have been tokenized, identifying paragraphs, sentences, and (word) tokens. In view of its size, the corpus has been tagged and lemmatized automatically, using Frog[8] (Van den Bosch et al., 2007). Unlike the Spoken Dutch Corpus, SoNaR came without exploitation software that would support users with limited or non-existent programming skills.

In the OpenSoNaR-CGN project, the texts of the Spoken Dutch Corpus or CGN have been curated and brought in line with the SoNaR-500 corpus by converting them to the FoLiA XML[9] format (van Gompel and Reynaert 2013).[10]

19.3 WhiteLab 2.0: Architecture

19.3.1 Design Considerations

Given the limitations in the functionality and scalability of existing tools, there clearly existed a great need for a new corpus exploitation suite in the Dutch language community. Since the development of COREX in 2003, technologies for web-based exploitation of large-scale datasets have also become more readily available and the use of these for linguistic research has been widely reported (Hoffmann and Evert, 2006; McEnery and Hardie, 2011; Hardie, 2012; Evert and Hardie, 2015). The need was partly met with the development of WhiteLab in the OpenSoNaR project (Reynaert et al., 2014). WhiteLab version 1.0 is a Java-based web application for the search and exploration of large-scale, linguistically annotated corpora. It caters to users of all skill levels by providing interfaces ranging from simple string querying to tools for advanced query composition, and even plain CQL entry using the Corpus Query Language, first introduced by Christ (1994). Metadata can be explored and queried in a comprehensive way. At the backend, WhiteLab relies on BlackLab and BlackLab-server[11] for corpus indexing and querying.

Nevertheless, the application was developed specifically for the SoNaR-500 corpus and, as such, does not provide support for speech-related annotations or audio. Furthermore, it can host only a single corpus, which limits its flexibility as a research tool. To overcome these issues, the

---

[8] In some of the data named identities have also been labeled.

[9] See Chapter 6 on FoLiA in this volume.

[10] As for the POS tagging, we observe that the tagset originally developed for tagging the Spoken Dutch Corpus Van Eynde et al. (2000) was later extended to account for tokens typically found in written texts Van Eynde (2005), and what was conceived as the CGN tagger-lemmatizer was reincarnated in Frog (`http://languagemachines.github.io/frog/`). Thus the POS tagging of CGN and SoNaR was already fully compatible.

[11] https://github.com/INL/BlackLab-server

OpenSoNaR-CGN project set out to develop WhiteLab 2.0 with the following considerations, which are in line with the recommendations made by Hoffmann and Evert (2006):

1. Users of different skill levels should be able to use the interface without problem, and continued use of the application should contribute to increasing a user's skill level.
2. The application needs to provide support for multiple corpora out of the box. Users should be able to query the corpora simultaneously, or separately.
3. The system should not be restricted to just the CGN and SoNaR-500 corpora, by providing support for widely used formats for content and metadata.
4. The manager of the application should have control over the metadata and how they are displayed in the interface. Since multiple corpora are now supported with multiple metadata formats, the manager should be able to group together fields with different labels under the same moniker in the interface.
5. Before querying the corpora, the user should be able to explore the data to get a sense of what is available.
6. Besides types, lemmata, and part-of-speech (POS) tags, phonetic transcriptions need to be indexed and made available for search.
7. Audio playback should be enabled for both recordings or parts thereof (hits).
8. All results should be exportable at least in CSV format for post-processing.
9. The application should be future-proof by investing in technologies that are particularly suited to the growing needs of the research community and are expected to stand the test of time to a reasonable degree.

Considering the previous version of WhiteLab, some of these criteria (1, 5, 8) had already been met in OpenSoNaR. The original application has been successfully applied in educational settings, proving its ability as a teaching tool. It also provides interfaces for both exploration and search, each with its own unique purpose and export functionality. Extensions made upon the interface are described in Section 19.4. Regarding the technical implementation, some choices have been made that really distinguish WhiteLab 2.0 from its predecessor, as we discuss in the remainder of this section.

19.3.2 System Design

A complete WhiteLab 2.0 setup consists of three components: an importer module to add corpora to the corpus index, a plugin to enable CQL searches on the index, and a web application that allows access to the index in an online context. For the first version of WhiteLab, the indexing and querying was handled by BlackLab and BlackLab-server. WhiteLab 2.0 also supports BlackLab, but by default it comes with its own newly developed WhiteLab 2.0 Importer and Plugin.

The most innovative aspect of WhiteLab 2.0 as opposed to WhiteLab is its use of the NoSQL graph database Neo4j (Neo Database AB, 2006). NoSQL databases have gained a lot of momentum over the last few years as a promising alternative to relational SQL databases for storing huge datasets. The main advantage of NoSQL over SQL[12] in general is its possibility to easily scale horizontally, meaning data may be spread over different servers, and its suitability for dynamic datasets. For the purpose of searching large collections of linguistically annotated data, two types of NoSQL databases are appropriate: document stores and graph databases. Document stores encapsulate data in structured documents, such as XML, which seems a perfect fit for linguistic corpora. However,

[12] Structured Query Language, standardized in ISO/IEC 9075-1:2011.

the specific structure of the FoLiA format that our corpora are encoded in makes it an arduous task to implement and optimise the arbitrary complex queries that can be produced by WhiteLab directly on the source documents. Therefore, a complete remapping of the data would likely be required when using a document store. Moreover, document stores are inherently document-centric, providing a strictly hierarchical view on the data.

Graph databases are similar to document stores, but incorporate the concept of relations between documents and other elements by modeling the data as a network. In contrast to document stores, this network is not necessarily hierarchical. It allows for references between (parts of) documents that would not be possible in a tree, which in turn provides a more expressive and easily navigable model of the data. Linguistic data encode networks of different natures, both syntactic and semantic, and both hierarchical and (seemingly) random, which would essentially be captured in a single database. Graph databases therefore seem a logical choice for our data and purpose, resulting in our choice for Neo4j. Neo4j stores data as nodes that are interconnected through relationships. Labels and properties may be defined on both nodes and relationships.

The web application itself has been redeveloped in Ruby on Rails (RoR).[13] RoR was chosen based on its transparent division of model, view, and controller, which increases speed of development and allows for easy extension of the application and reuse of its parts in other applications.

19.3.3 Data Model

Figure 19.1 displays the WhiteLab 2.0 data model implemented in Neo4j for linguistically annotated corpora. When designing a data model for Neo4j, it is important to consider the sizes of nodes, relationships, and their properties as they are stored in the database, which are respectively 15, 34, and 64 bytes.[14] A combination of two nodes and a relationship requires less storage space than a single node with one property (64 versus 79 bytes). Therefore, it is always more efficient to store an element attribute as a new node rather than a property, and connect the element node to it (Figure 19.2, red and black lines). In this scenario, the extra property node would not require its own property, since Neo4j allows nodes to be identified or typed using labels. However, at the time of development of WhiteLab 2.0, it was not possible to efficiently query labels using regular expressions, which is a base requirement for the target audience. Properties can be indexed for improved

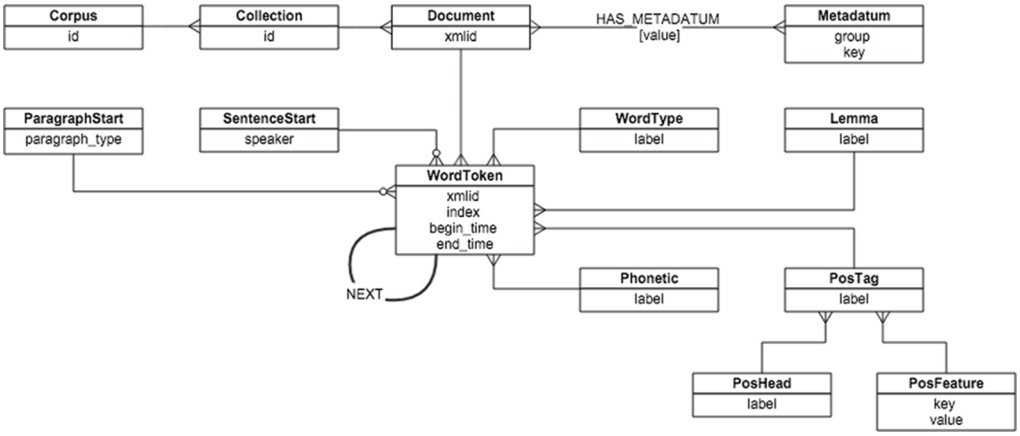

Figure 19.1: The WhiteLab 2.0 data model for the Neo4j backend.

[13] http://api.rubyonrails.org/

[14] http://neo4j.com/docs/stable/configuration-io-examples.html

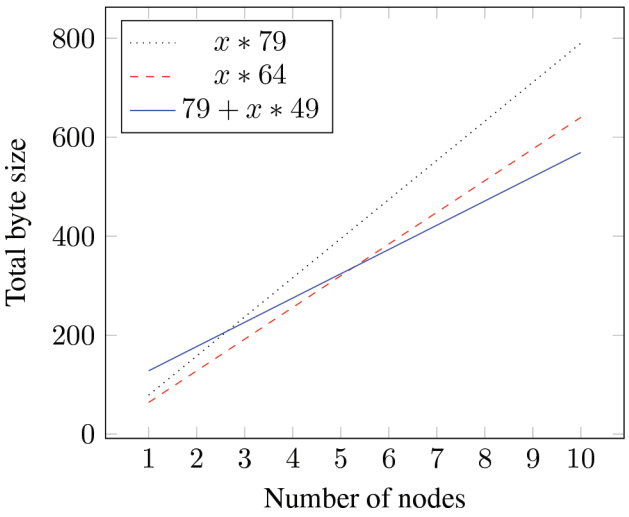

Figure 19.2: The WhiteLab 2.0 data model for the Neo4j backend.

search and do allow for regular expression queries, but the total size of two nodes, a relationship and one property (128 bytes) is larger than a single node with one property. Nevertheless, if more elements have the same attribute and their nodes connect to the same property node, the total size quickly becomes less than when storing each element node with its own property (Figure 19.2, blue, full line). In practice this means that attribute values with a frequency lower than 3 are most efficiently stored as properties of the element node they describe, whereas higher frequency attribute values should be placed in their own node, which is then connected to the appropriate element nodes.

Compared to WhiteLab, the set of annotations in WhiteLab 2.0 has been extended to include phonetics. This includes the addition of token attributes regarding the token's position in the audio, as well as identification of the speaker at sentence level. Also, the Part-of-Speech attributes (head and features) are separated from the complete tag, allowing for more fine-grained analysis of these annotations.

In order to retain support for BlackLab, a new index tool has been added to BlackLab[15] that enables indexing of multiple corpora, and a set of new BlackLab indexers has been developed specifically for use with WhiteLab 2.0. With the current indexers and Importer, the size of the Neo4j database is approximately twice that of the equivalent BlackLab index.

19.3.4 Administration

A new feature in WhiteLab 2.0 is the Admin interface. It allows the application manager to inspect and manage the metadata and Part-of-Speech tags across corpora. This functionality was added in light of known differences in the tagsets used for SoNaR-500 and CGN, which can now be easily inspected. The Admin interface also allows control over the interface language and info page content (Section 19.4).

The types of corpora that WhiteLab 2.0 is designed to make accessible are mostly of a static nature, certainly so at the document level. Therefore, result sets for queries are not expected to change after deployment. We have taken advantage of this fact by including an SQL database in the web application for user and query logging. The query logging is set up in such a way that no

[15] https://github.com/Taalmonsters/BlackLab

duplicate queries are sent to the Neo4j database. For instance, if two users enter the same query within a short timeframe, the query is sent to the database only upon first request. The second request will simply wait for the first to finish and then access its results. Another request for the same data at a later time will also quickly return the previously stored result to the user. To keep a handle on the resources used, the web application includes some easy-to-set-up scheduled tasks, so-called Cron jobs, that run daily to remove queries that have timed out or have not been accessed in a while. A further advantage are the insights that the application manager may gather from the query statistics.

19.3.5 Performance

We test the performance of the WhiteLab 2.0 plugin for Neo4j compared to that of BlackLab-server 1.3 on the same dataset. Due to limitations of available hardware, the tests are performed on a subset of the complete OpenSoNaR+ data, namely, the entire CGN, plus the following SoNaR-500 collections: WR-P-E-E Newsletters, WR-P-E-F Press releases, WR-P-E-H Teletext pages, WR-P-E-J Wikipedia, WR-P-E-K Blogs, WR-P-P-B Books, WR-P-P-D Newsletters, WR-P-P-I Policy documents, WR-P-P-K Reports, and WR-U-E-A Chats[16]. The total size is around 83 million tokens. Tests are performed in a single dedicated server setup on a 12-core system with 64 Gb of RAM.

We test the response time of both backends to five queries with increasing absolute hit counts ranging from approximately 7,500 to 250,000 hits. Each query is sent to the server 51 times over the command line. By bypassing the GUI, we disable the WhiteLab query caching for these tests. The first call is discarded for both backends, as this warms up the index and takes considerably more time to complete. Figure 19.3 shows the average response time over the remaining 50 calls for each query. As is shown, BlackLab's performance is unhindered by increasing hit counts, where WhiteLab 2.0's response time increases almost linearly to the hit count. When we inspect the logs, we see that Neo4j spends most time on collecting the nodes that match the first token in the CQL query, and on grouping results where necessary. Similar tests on queries of increasing complexity

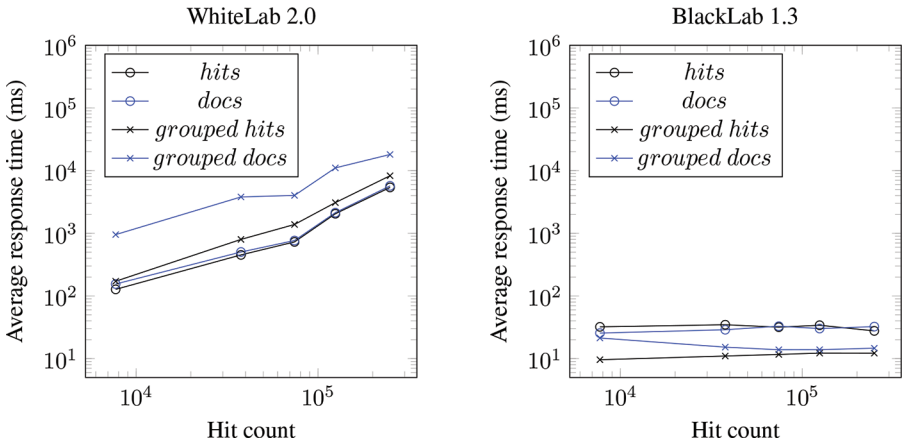

Figure 19.3: Performance of WhiteLab 2.0 compared to BlackLab 1.3. Five queries with increasing result counts (in absolute hits) are each performed 51 times. The first call is discarded, as this warms up the indexes. We report the average response time over the remaining 50 calls.

[16] The SoNaR User Manual (available from `http://ticclops.uvt.nl/SoNaR_end-user_documentation_v.1.0.4.pdf`) explains these codes.

as measured in n-gram size confirm that the initial node selection is the crux; the n-gram size has little to no effect on the response time. The delay in the grouping is particularly detrimental to queries for grouped documents, as these require two groupings: first from hits to documents, and then into groups of documents. Overall, the tests show that there is still a lot to be done in terms of optimization of the WhiteLab 2.0 Neo4j plugin.

19.4 OpenSoNaR+: User Functionality[17]

We describe the WhiteLab 2.0 web interface as it is designed for OpenSoNaR+. It largely resembles the WhiteLab interface for OpenSoNaR, with added support for audio. A major advantage over the previous version is the addition of easy-to-configure interface translations. By default, the application comes with Dutch and English translations, but these may be extended by the application manager through the Admin interface. This interface also provides functionality to streamline metadata over different corpora. The metadata labels and values are listed including their coverage of indexed corpora, which provides a quick overview of possible similarities and discrepancies between corpus metadata. Different labels that refer to the same type of information can be grouped together under the same label. The translation functionality used for the interface components is also applied to the metadata labels.

The WhiteLab user by default lands on the **Search** page of OpenSoNaR+ when logging in. Next to this page we have the **Explore** page and the **Info** page. The Admin interface is hidden behind a login page and thus not available to regular users.

19.4.1 Info Page

The Info page provides information about the system. It provides a first-user manual which gives an overview of the main functionalities that OpenSoNaR+ offers. It also provides the user manuals of the SoNaR and CGN corpora, which offer in-depth information on the composition of both the contemporary written Dutch corpus and the spoken Dutch corpus. The system also provides a guided tour to its users, which gives the user a quick introduction to each page's uses and possibilities. Access to the guided tour is through the question mark button to the left in the top bar of the interface.

19.4.2 Explore

The Explore page gives statistical and visual information about the corpus contents. It provides insight into the distribution of the texts available per genre and according to their provenance, which is basically whether they were collected in the Netherlands or in Flanders or are of unidentified or unidentifiable provenance. The latter is the case for example with text materials obtained from the European Union or Wikipedia.

On the basis of metadata selections under the 'statistics' tab, the user can obtain custom frequency lists for particular subselections of the corpora. These are further discussed under Subsection 19.4.5. This page also affords access to *n*-gram (where *n* is 1 to 5) frequency lists derived from subcorpora for word forms, lemmata, POS tags and phonetic transcriptions.

Finally, the page affords direct access to a particular document in the incorporated collections on the basis of its file name. This should be a useful feature for possible research verification or replication when the particular document has been referred to in a research paper.

[17] This section is an adaptation and extension of the initial description in Reynaert et al. 2014.

19.4.3 Search

The Search environment is to date the most elaborate. It provides four levels of access to the contents: Simple, Extended, Advanced and Expert.

The **Simple search** option provides Google-style, single query box access. Entering a search term here will instantiate a search over the full contents of the corpus. The search is for word forms, which may be phrases (*n*-grams), in which case insensitive matches are sought that respect the actual sequence of words. This latter functionality is also provided by the Extended and Advanced search environments.

The **Extended search** environment allows one to impose selection filters on the search effected. These filters are of two kinds. First, there are filters on the metadata. Second, there are filters on the lexical level, allowing one to search for either word forms, lemmata, POS tags and phonetic transcriptions for the spoken Dutch data.

The metadata filters are at first hidden behind a bar visible above the actual lexical query fields. When the user wants to impose metadata filters the bar is expanded by a simple mouse click and the user is presented with a row consisting of three drop-down boxes. The middle box has just two options: 'is' or 'is not'. The left box gives access to all the metadata fields available in the corpus CMDI metadata files. The right box, upon selection of a particular metadata field in the left box, dynamically expands with the list of available metadata contents, where applicable. Metadata filters can be stacked. Through a 'plus' button to the right of the query row, one may obtain further rows in each of which further restrictions on the query may be imposed. The metadata view shows the proportional and absolute (i.e. number of tokens) size of the dataset matching the currently selected filters. When a metadata filter is selected or updated, these numbers are automatically updated, allowing the user to quickly inspect subcorpus size prior to searching.

The metadata selection interface additionally provides the option of grouping the query results obtained by a range of features. For example, if one here selects the option of having the results presented by country of origin of the hit texts, one is not presented directly with the Key Words in Context (KWIC) list of results, but rather with a bar representation of the number of hits per country. One may then click on one of these bars and be presented with the KWIC list. This then gives the user the possibility to select one of these subsets and to further work on these as a new, independent query.

The lexical filters allow one to optionally perform case-sensitive searches for word forms, lemmata and/or phonetic transcriptions. POS tags can of course be searched too. When the search is for lemmata, all the word forms sharing the same lemma will be retrieved. For POS tag searches the user is presented with a drop-down list which presents a layperson's translation in plain language for the actual POS tags involved. Combinations of, for instance, word forms and POS searches are possible to direct the search for the word 'drink' (ibidem in English) towards the first person singular of the present tense verb form, rather than its use as a noun.

For the **Advanced search** option we fully acknowledge to have emulated the elegant interface to CQL-query building as provided by the Swedish Språkbanken[18]. Users are first presented with a single box containing three query fields. By horizontally or vertically adding further boxes they may build quite complex queries without the need to know the query language behind them. Vertical boxes may be stacked with 'and' or 'or' conditions. These boxes give access not only to queries on full word forms (word 'is' or word 'is not') but also to words beginning with or containing or ending with a specific character string. Regular expressions are a further option. Users get to see the query they have built and have the option of further extending it, manually.

[18] See 'Korp' at `https://spraakbanken.gu.se/korp/#?lang=en&search_tab=1`

The **Expert search** requires knowledge of the query language incorporated in the system. It is CQL, the Corpus Query Language[19]. In its essence, this search option's limitations are defined mainly by the user's CQL proficiency. However, to support the educational requirements of WhiteLab 2.0, queries can be entered in one interface (e.g. Simple search) and viewed in another, more complex interface (e.g. Expert search) without first having to execute the query. Using this functionality, students and laypeople can directly see the CQL query generated from their string query and actually increase their familiarity with the Corpus Query Language.

19.4.4 Presentation of the Results

Regardless of the search option one has chosen, by default, eventually a KWIC list of results is presented. A red button for each of the text snippets gives direct access to the full-text view of the document. There, moving the cursor over any of the words in the text, one gets to see a small window with the word form's unique ID, lemma and POS tag. Documents retrieved from CGN have a button for the whole text and buttons per sentence for calling up the appropriate sound recording. New tabs give access to the particular document's full metadata, to document specific statistics on size in terms of word tokens and types and derived measures. Finally, the user is presented with a visualisation of the token to POS tag distribution and the vocabulary growth curve.

A feature of the Extended and Advanced search options we have not seen in other corpus exploration environments is that multiple queries can be performed in one operation. This is facilitated by the fact that by clicking on the 'list' button to the right of the query boxes the user may effortlessly upload a pre-prepared list of query terms. After uploading, these query terms are converted by the system into actual, separate CQL queries which are visible in the query history. The user then has the option of having the output presented separately, per query, or mixed. If in the Advanced search environment a user uploads more than one query list, the system makes a combination of all the query terms in the lists. Given x terms in list A and y terms in list B, this results in x times y queries. If this is not what the user intended, then the user has the option of uploading a list of, for instance, word bigrams to be searched for in the Extended search environment.

19.4.5 Export of the Results

Both the Explore and Search pages allow the users to export the results of their queries. This would be the frequency list built on the basis of the selections made, whether metadata-based, lexical, or indeed both. Or else, one may export the list of documents that were selected. What WhiteLab by design does not provide, is export of the full documents. This facility exceeds for the best part the IPR-agreements that were achieved with the text providers. However, the full corpora containing the full texts are freely obtainable for research purposes from the INT.

The query results are exported in various formats, including comma-separated lists suitable for loading in a spreadsheet. The format should be easily convertible to the specific formats required by statistical packages such as R[20] or SPSS[21].

19.4.6 Query History

An important new feature of the updated WhiteLab is that a user's query history is stored and is accessible to the user through an unobtrusive sea-green button in the lower left corner of the window.

---

[19] A nice tutorial is at: `http://cwb.sourceforge.net/files/CQP_Tutorial/`.

[20] `https://www.r-project.org/`

[21] `https://www.ibm.com/marketplace/cloud/statistical-analysis-and-reporting/`

The results of one's export actions are to be found here as part of the summary of each query one has undertaken.

19.5 Performance and Availability

19.5.1 Performance

As far as technologies go, Neo4j is relatively young and still in active development. Since the start of the OpenSoNaR-CGN project, many new versions have already been released, including updates that will likely increase performance for WhiteLab 2.0 once implemented. This trend is expected to continue over the coming years. WhiteLab 2.0 is already set up to reap the benefits of these advancements, while also being able to function with established technologies through BlackLab.

Currently, the query caching of the interface is resolved using an SQL database. We recognize that this can also be solved using a key-value store such as Redis. At the time of development of the current version of WhiteLab, use of Redis still imposed a lot of security risks and was not advised in production systems. However, recent developments have greatly improved Redis's security[22], making it a feasible alternative that we will definitely consider. Moreover, BlackLab-server provides its own query caching.

Independent from external developments, we see a number of possibilities for improving performance of the Neo4j backend. Certainly, the application itself can be further optimized and streamlined. But most benefit would likely be gained from decreasing the size of the Neo4j database, either through simplification of the data model, or separation of structure and content. The latter could be achieved, for instance, through a dual-database setup, where one database holds the document structures and the other the linguistic network. Another possibility we intend to investigate is storage of the annotations in an optimized string index such as the one used by word2vec (Mikolov et al., 2013), which reaches great speeds on huge collections of strings.

19.5.2 Availability

All WhiteLab 2.0 components are released under the GNU Affero General Public License and are currently available at `https://github.com/Taalmonsters/WhiteLab2.0`. An installation manual for use with either the BlackLab or the Neo4j backend is provided.

19.6 Conclusion

The distribution version of CGN requires 115GB in archived form and SoNaR-500 takes up 62.6GB. Unwieldy at best, and to all intents and purposes practically inaccessible to the average researcher. Though freely available for research, some unwary researchers were nastily surprised when trying to unpack SoNaR on common laptops running everyday software. Reports of these mishaps prompted the original OpenSoNaR project proposal to be written.

Results of the first project being well-received, OpenSoNaR-CGN followed suit. In relatively little time and on a modest budget with a small, but dedicated, team, we have managed to put OpenSoNaR+ – both corpora, text and sound – at everyone's fingertips.

We hope WhiteLab may serve researchers well. We definitely hope it will find favour with new and existing corpus endeavours in the Low Countries and far beyond.

[22] `https://www.reddit.com/r/redis/comments/3zv85m/new_security_feature_redis_protected_mode/`

Acknowledgements

We gratefully acknowledge the feedback we received from our user group and the funding provided by CLARIN-NL under grant numbers CLARIN-NL-12-013 and CLARIN-NL-15-005. Martin Reynaert further acknowledges being funded by the new Dutch national CLARIN project CLARIAH and by NWO in project Nederlab. Finally we would like to thank the two anonymous reviewers of the prefinal version of this chapter for their constructive feedback.

References

Christ, Oliver (1994) A modular and flexible architecture for an integrated corpus query system. *Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research*. Budapest, Hungary. pp. 23–32.

Elisabeth D'Halleweyn, Jan Odijk, Lisanne Teunissen, and Catia Cucchiarini. 2006. The Dutch-Flemish HLT Programme STEVIN: Essential Speech and Language Technology Resources. In Nicoletta Calzolari et al., editor, *Proceedings of the Fifth international conference on Language Resources and Evaluation (LREC-2006)*, pages 761–766, Genoa, Italy. European Language Resources Association (ELRA).

Stefan Evert and Andrew Hardie. 2015. Ziggurat: A new data model and indexing format for large annotated text corpora. In Piotr Bański, Hanno Biber, Evelyn Breiteneder, Marc Kupietz, Harald Lüngen, and Andreas Witt, editors, *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3), Lancaster, 20 July 2015*, pages 21–27.

Andrew Hardie. 2012. CQPweb - combining power, flexibility and usability in a corpus analysis tool. *International Journal of Corpus Linguistics*, 17(3):380–409.

Sebastian Hoffmann and Stefan Evert, 2006. *BNCweb (CQP edition) - the marriage of two corpus tools*, pages 177–195. Peter Lang.

Tony McEnery and Andrew Hardie. 2011. *Corpus Linguistics: Method, Theory and Practice*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Neo Database AB. 2006. The Neo Database – A Technology Introduction. http://dist.neo4j.org/neo-technology-introduction.pdf.

Jan Odijk. 2010. The CLARIN-NL project. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC-2010*, pages 48–53, Valletta, Malta.

Nelleke Oostdijk and Daan Broeder. 2003. The Spoken Dutch Corpus and its exploitation environment. In A. Abeille, S. Hansen-Schirra, and H. Uszkoreit, editors, *Proceedings of the 4th International Workshop on linguistically interpreted corpora (LINC-03)*, pages 93–101.

Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman. 2013. The construction of a 500-million-word reference corpus of contemporary written Dutch. In *Essential Speech and Language Technology for Dutch: Results by the STEVIN-programme*, chapter 13, pages 219–247. Springer Verlag.

Nelleke Oostdijk. 2000. The Spoken Dutch Corpus. Overview and first evaluation. In Nicoletta Calzolari et al., editor, *Proceedings of the Second international conference on Language Resources and Evaluation (LREC-2000)*, pages 887–894, Athens, Greece. European Language Resources Association (ELRA).

Martin Reynaert, Matje van de Camp, and Menno van Zaanen. 2014. OpenSoNaR: user-driven development of the SoNaR corpus interfaces. In *Proceedings of COLING 2014: System*

Demonstrations, pages 124–128, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Peter Spyns and Jan Odijk, editors. 2013. *Essential Speech and Language Technology for Dutch. Results by the STEVIN-programme*. Theory and Applications of Natural Language Processing. Springer-Verlag, Berlin.

Helmer Strik, Walter Daelemans, Diana Binnenpoorte, Janienke Sturm, Folkert De Vriend, and Catia Cucchiarini. 2002. Dutch HLT resources: from BLARK to priority lists. In *Proceedings of ICSLP-2002*, pages 1549–1552, Denver.

Antal Van den Bosch, Gertjan Busser, Sander Canisius, and Walter Daelemans. 2007. An efficient memory-based morpho-syntactic tagger and parser for Dutch. In P. Dirix et al., editor, *Computational Linguistics in the Netherlands: Selected Papers from the Seventeenth CLIN Meeting*, pages 99–114, Leuven, Belgium.

Frank Van Eynde, Jakub Zavrel, and Walter Daelemans. 2000. Part of speech tagging and lemmatisation for the Spoken Dutch Corpus. In *Proceedings of the Second international conference on Language Resources and Evaluation (LREC-2000)*, pages 1427–1433, Athens, Greece.

Frank Van Eynde. 2005. Part of speech tagging en lemmatisering. Protocol voor annotatoren in D-Coi. Technical report, Centrum voor Computerlinguïstiek, K.U. Leuven.

Maarten van Gompel and Martin Reynaert. 2013. FoLiA: A practical XML Format for Linguistic Annotation - a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3.