Challenges in the Management of Large Corpora

Workshop Programme

14:00 – Opening

14.03 – 14.30 Keynote talk
Nancy Ide,
Big, Clean, and Comprehensive – but is it Worth it?

14.30 – 15.00
Lars Bungum and Björn Gambäck,
Efficient N-gram Language Modeling for Billion Word Web Corpora

15.00 – 15.30
Hans Martin Lehmann and Gerold Schneider,
Dependency Bank

15.30 – 16.00
Roman Schneider,
Evaluating DBMS-based Access Strategies to Very Large Multi-layer Corpora

16:00 – 16:30 Coffee break

16.30 – 17.00
Hanno Biber and Evelyn Breiteneder,
The AAC Container. Managing Text Resources for Text Studies

17.00 – 17.30
Damir Ćavar, Helen Aristar-Dry and Anthony Aristar,
Large Mailing List Corpora: Management, Annotation and Repository

17.30 – 18.00
Ritesh Kumar, Pinkey Nainwani, Girish Nath Jha and Shiv Bhusan Kaushik,
Creating and Managing Large Annotated Parallel Corpora of Indian Languages

18.00 – 18.30
Nelleke Oostdijk and Henk van den Heuvel,
Introducing the CLARIN-NL Data Curation Service

18.30 – 19.00 Final discussion

Editors

Piotr Bański	Institut für Deutsche Sprache, Mannheim
Marc Kupietz	Institut für Deutsche Sprache, Mannheim
Andreas Witt	Institut für Deutsche Sprache, Mannheim
Damir Ćavar	Institute for Language Information and Technology, Eastern Michigan University
Serge Heiden	ICAR Laboratory, Lyon University
Anthony Aristar	Institute for Language Information and Technology, Eastern Michigan University
Helen Aristar-Dry	Institute for Language Information and Technology, Eastern Michigan University

Organizing Committee

Anthony Aristar	Institute for Language Information and Technology, Eastern Michigan University
Helen Aristar-Dry	Institute for Language Information and Technology, Eastern Michigan University
Piotr Bański	Institut für Deutsche Sprache, Mannheim
Damir Ćavar	Institute for Language Information and Technology, Eastern Michigan University
Serge Heiden	ICAR Laboratory, Lyon University
Marc Kupietz	Institut für Deutsche Sprache, Mannheim
Andreas Witt	Institut für Deutsche Sprache, Mannheim

Workshop Programme Committee

Núria Bel	Universitat Pompeu Fabra
Mark Davies	Brigham Young University
Stefanie Dipper	Ruhr-Universität Bochum
Tomaž Erjavec	Jožef Stefan Institute
Stefan Evert	Technische Universität Darmstadt
Alexander Geyken	Berlin-Brandenburgische Akademie der Wissenschaften
Andrew Hardie	University of Lancaster
Nancy Ide	Vassar College
Sandra Kübler	Indiana University
Martin Mueller	Northwestern University
Mark Olsen	University of Chicago
Adam Przepiórkowski	Polish Academy of Sciences, University of Warsaw
Reinhard Rapp	Johannes Gutenberg-Universität Mainz, University of Leeds
Laurent Romary	INRIA, Humboldt-Universität zu Berlin
Pavel Straňák	Charles University in Prague
Amir Zeldes	Humboldt-Universität zu Berlin

Table of contents

Author Index

Introduction

We live in an age where the well-known maxim that "the only thing better than data is more data" is something that no longer sets unattainable goals. Creating extremely large corpora is no longer a challenge, given the proven methods that lie behind e.g. applying the Web-as-Corpus approach or utilizing Google's n-gram collection. Indeed, the challenge is now shifted towards dealing with the large amounts of primary data and much larger amounts of annotation data. On the one hand, this challenge concerns finding new (corpus-) linguistic methodologies that can make use of such extremely large corpora e.g. in order to investigate rare phenomena involving multiple lexical items or to find and represent fine-grained sub-regularities; on the other hand, some fundamental technical methods and strategies are being called into question. These include e.g. successful curation of the data, management of collections that span multiple volumes or that are distributed across several centres, methods to clean the data from non-linguistic intrusions or duplicates, as well as automatic annotation methods or innovative corpus architectures that maximise the usefulness of data or allow to search and to analyze it efficiently. Among the new tasks are also collaborative manual annotation and methods to manage it as well as new challenges to the statistical analysis of such data and metadata.

The workshop on "Challenges in the management of large corpora" aims at gathering the leading researchers in the field of Language Resource creation and Corpus Linguistics, in order to provide for an intensive exchange of expertise, results and ideas.

Big, Clean, and Comprehensive–But is it Worth It?

Nancy Ide

Department of Computer Science
Vassar College, USA
ide@cs.vassar.edu

Abstract

Several projects have devoted considerable time, effort, and funding to the development of language corpora, in order to provide large amounts of linguistically annotated data to support natural language processing research and development, and in particular for developing statistical language models that can enable machine learning. As opposed to data collected from the web, these corpora are ”clean” (In the sense of having been rendered into a tractable format for processing), can be enhanced with multiple layers of linguistic annotation, can be designed to cover a “representative” set of genres, and, perhaps most importantly, can be re-distributed for reuse by others–clear advantages that on the face of it seem to justify the effort of constructing these corpora. However, corpus construction is only a first step; to be of real use, the data and annotations must be searchable and accessible via methods that go well beyond simple ”Google search”, thus potentially demanding software development by institutions with limited personnel and funding. Beyond this is the effort required to maintain the corpus and the software and provide for their access and distribution, which in itself can demand a major investment of time and resources. We can even consider efforts to develop standards that might contribute to data and software reuse as another significant cost of large corpus development. This talk will attempt to weigh the benefits that these resources provide to the natural language processing and linguistics communities against the time, effort, and expense of language resource development, in order to determine whether or not the benefits justify the costs. I will look at the uses to which language corpora are put by these communities and consider the degree to which carefully-constructed, annotated language corpora enable research and development that is *quantifiably* beyond what could be done using web resources–either existing resources or what we can assume is in the foreseeable future. I will also consider the likelihood that, given their far superior resources, enterprises such as Google and/or projects such as the Semantic Web will eventually render large corpus construction and maintenance unnecessary.

Keywords: Large corpora, Linguistic corpus use, Corpus maintenance

The AAC Container. Managing Text Resources for Text Studies.

Hanno Biber, Evelyn Breiteneder
Institute for Corpus Linguistics and Text Technology, Austrian Academy of Sciences
Sonnenfelsgasse 19/8, 1010 Wien
E-mail: hanno.biber@oeaw.ac.at, evelyn.breiteneder@oeaw.ac.at

Abstract

The aim of this paper about the concept of the "AAC container" is to contribute to the workshop theme of managing large corpora by putting emphasis on the perspective of how to come to terms with the actual content of a text corpus by applying approaches based upon the methodologies of text studies. The "AAC-Austrian Academy Corpus" is a large digital text corpus operated by the "Institute for Corpus Linguistics and Text Technology" of the "Austrian Academy of Sciences" in Vienna. Thousands of German language documents and literary objects by thousands of authors have been collected. The historical period covered by this text corpus of 500 million tokens is ranging from the 1848 revolution to the fall of the iron curtain in 1989. In this period significant historical changes with remarkable influences on the language and the language use can be observed. Among the AAC's sources, which cover many domains and genres, there are literary journals, newspapers, novels, dramas, poems, advertisements, essays, travel accounts, cookbooks, pamphlets, political speeches, scientific, legal, religious texts, etc. The AAC corpus holdings provide a great number of reliable resources and interesting corpus based approaches for investigations into the linguistic and textual properties of these texts.

Keywords: text corpora, literary studies, corpus linguistics

1. The AAC Container

Figure 1: AAC Container

1.1. AAC-Austrian Academy Corpus

The aim of this paper is to contribute to the workshop theme of managing large corpora by putting a particular operational emphasis on the perspective of how to come to terms with the actual content of a large text corpus by applying approaches based upon the methodologies of text studies. The "AAC-Austrian Academy Corpus" has been developed for these purposes several years ago and constitutes a large digital text corpus. It is now operated by the "Institute for Corpus Linguistics and Text Technology" of the "Austrian Academy of Sciences" in Vienna. Thousands of primarily German language texts, documents and literary objects of considerable historical and linguistic significance, written by thousands of authors have been collected.

1.2. A Historical Text Corpus from 1848 to 1989

The historical period covered by the corpus is ranging from the 1848 revolution to the fall of the iron curtain in 1989. In this period significant historical changes with remarkable influences on the German language and in particular the specific language use can be observed. Among the AAC's sources, which cover many different domains and genres, there are literary journals, magazines, newspapers, novels, dramas, poems, advertisements, essays, travel accounts and travel guides, cookbooks, pamphlets, political brochures and political speeches, scientific texts from various fields, legal documents, religious texts, military documents, schoolbooks, yearbooks, almanacs, special collections, comic literature, art magazines, avant-garde and modernist literature, and also translated literature of cultural significance etc. The AAC corpus holdings provide a great number of highly reliable resources and interesting corpus based research approaches for investigations into the linguistic and textual properties of these texts.

1.3. 500.000.000 Tokens

More than 500 million tokens or to be more precise around 500 million running words of text have already been scanned and converted into machine-readable text. In very many cases these digital texts have been very carefully annotated and basic structural mark-up and selected thematic mark-up has been applied according to annotation and mark-up schemes based upon XML related standards, whereby the corpus researchers have been working on various issues of digitization of historical language data, establishing professional workflows as well as developing practical software in order to be able to deal with the large amount of information and data processing.

1.4. Text Resources for Text Studies

1.4.1. Collecting Texts

While the objectives of the build-up phase of the corpus starting in the early 2000s were focused on issues of corpus creation based upon principles of collecting texts of crucial historical, cultural and linguistic significance, the next phase will be focused on more research on the literary and historical analysis and exploitation of these vast textual resources.

1.4.2. Text Documentation

It is first and foremost necessary to describe and document the corpus resources in a way that enables users from different backgrounds and disciplines to do research with the text documents. The systematic documentation and description of the sources is of particular interest for historical text studies.

1.4.3. Text Analysis

In general, corpus based text analytical approaches will be at the core of the research activities, so that not only special research interests but also broader research approaches towards the textual and structural properties of a great variety of literary documents or of documents from a wider area of publication can be followed.

1.4.4. Corpus Methodologies

The main work will be to adapt existing resources to the needs for scholarly text analysis based upon the principles and methodologies of corpus research and corpus linguistics. Tools and applications that are modelled according to the principles of corpus linguistics will help to achieve better results in this process.

1.4.5. AAC Literary Journals

Among many research fields and possible research approaches and directions for the study of historical literary sources the large amount of full runs of historical literary journals and modernist magazines within the AAC, the large amount of such important sources of cultural production present in the corpus is of particular interest here. This is of crucial scholarly interest because the literary journals are to be considered as a primary and most important publication platform for writers at the time of the corpus, particularly so in the first half of the 20th century. There would not have been the literary life of the time as it was without these specific publication instruments.

1.4.6. AAC-Fackel

By 2007 the "AAC-Fackel", a digital online edition of "Die Fackel" by Karl Kraus has been established and published within the framework of the "AAC-Austrian Academy Corpus" in Vienna. This work, like the similar edition of "Brenner online" is an example case of a digital text resource in the specific format of a scholarly digital edition.

1.4.7. "Die Fackel"

The original journal "Die Fackel" constitutes an important work of world literature and is one of the most significant literary sources of the German language of this time. It was originally published and almost entirely self-authored by the famous satirist Karl Kraus in Vienna from 1899 until 1936. As a satirical writer, as a language critic and as a social critic Karl Kraus observed very carefully and critically comments upon the language used in the newspapers and the publications of his time. His main method critique is the satirical method of quotation, by means of which he exposes the failures and crimes of his contemporaries. In numerous texts he satirically glosses the words and phrases of others. He is - in one of his highly productive periods as a satirical writer - treating the atrocities of the First World War and with particular emphasis the linguistic behaviour that led to the bloodshed as well as – later - to the development of an even more monstrous tragedy.

1.4.8. 1933

In addition to the "AAC-Fackel" new resources will be developed, if possible, with a special emphasis of the historical period around the year 1933. The topic of this research sub-project is focused on the questions of developing a diachronic text corpus of historical significance and the establishment of a corpus based research environment for language studies of the interwar period focusing the year 1933, the year when the NSDAP came to power in Germany.

1.4.9. 1923-1938

The year 1933 and the years preceding and following the seizure of power of the National socialists is a historical period of particular interest for language studies. In this case not primarily the well-known documents and the evident language will be included in the analysis, but systematically the less easily visible documents and less significant lexical items could be taken into consideration.

1.4.10. Studying Texts

The AAC collection principles have been, to a large extent – besides other more traditional corpus linguistic parameters of text selection - been deliberately determined upon a selection process that has been guided by the work of Karl Kraus, who has to be regarded as an exact observer and critic of the language of his time, a corpus selection process which has been termed "Fackel-induced" thus providing the researchers with a critical instrument. The methodological approach of providing text resources for text studies is considered as particularly fruitful by means of applying methods of corpus linguistics and by testing new strategies of the application of these methods in the context of historical language studies.

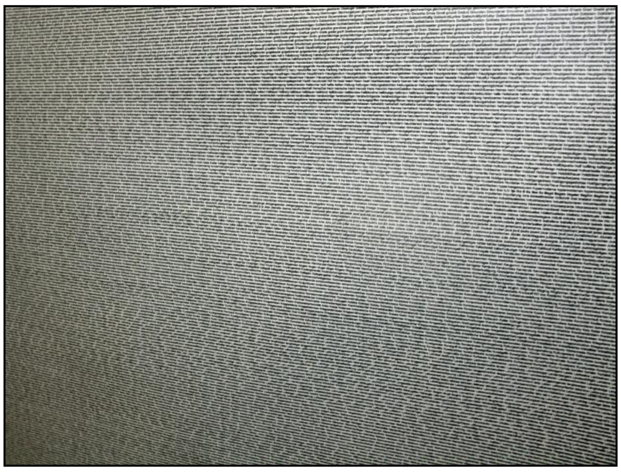


Figure 2: AAC Tokens

1.5. Corpus Building Objectives

The AAC has for several years already made use of text-oriented concepts thereby attaching great importance to a particular perspective that does not allow the simple reduction of such resources to mere collections of linguistic items, words, sentences and the like. Without appropriate tools and analytical instruments that provide the literary scholars as well as the linguists as users of large language resources and text corpora with structured access to and correct information about these text documents, in which the language data is contained, valuable knowledge about and interesting insights into these resources will be questionable and problematic.

The corpus research will have to focus on methods and resources for making large amounts of texts accessible in a well-structured way, so that interpretation of the texts can be possible. Efforts are made to develop usable tools, attempting to add to them wherever necessary, to provide relevant expertise while building up what we call the AAC Container.

1.6. Corpus-based Literary Studies

The AAC Container is a systematic central and well-structured access structure to the holdings of the entire corpus. Corpus linguistics and corpus research and the creation of large electronic text collections have traditionally been the domain of linguists. Literary digitization initiatives were often restricted to particular writers, and many of these scholarly projects did neither produce large amounts of data nor pursue research on methods of how to cope with the considerable problems involved in working with such data. Being aware of the need of digital resources in many fields of the humanities, the AAC has started to work on methods as well as tools and specific applications with a wider scope as far as the need for research instruments and applications for textual studies is concerned.

While the needs of linguists have not been ignored, this approach of text-oriented computing aimed at solutions that also offered access to coherent texts, being convinced that for many applications it is essential for researchers to have as direct access to the texts as possible. From the start, the AAC relied on XML as the foundation of their corpus build-up activities and the AAC tools support this technology allowing both the controlled application of mark-up as well as automated validation of large amounts of data.

1.7. AAC Encoding

The AAC used to apply an encoding scheme characterized by a combined approach to capture structural features of the texts and at the same time a certain amount of data describing the physical appearance of the original texts. This approach has led to a system of mark-up not only representing the basic semantic structures of the texts but also some amount of layout information. In digitizing historical data, semantic and presentational data will remain together. When working on large amounts of such texts, it is easier to capture formal data than to translate typographic idiosyncrasies into consistent structural mark-up.

The AAC's digitizing activities in the first years of its development have been characterized by a strong connection to the physical objects of digitization. The AAC attaches importance to the semantic structures of the text as well as to the physical appearance of the text. The output is visualized via XSLT in browsers and encoding tools. It contains precise specifications and explanations of the elements and attributes that make up the system, and gives numerous examples intended to help users to correctly apply mark-up.

1.8. AAC Metadata

Metadata describing production processes and details about the physical sources of the digitized object are the back-bone of a digital data collection's usability. The AAC collects two types of data that fall into these categories. The first consists in descriptive metadata concerning the digitized objects. This information has been drawn-up on a regular basis when the physical objects were scanned. It is stored in a relational database containing more than 6000 records holding all relevant information about the physical print objects so far incorporated into the corpus. In many cases this data is much more detailed than regular library records. The record fields in this database were designed in a way that they can be easily mapped onto the fields of TEI headers. The process of converting the corpus data structures into TEI compatible formats is currently performed by a special task force at the "Institute of Corpus Linguistics" and will be finished by the end of 2012 and the results of this process will be and have to some extent been presented separately.

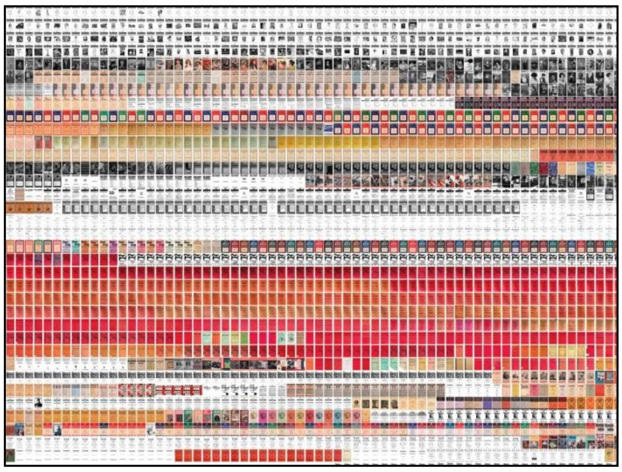


Figure 3: AAC Magazines

1.9. Corpus Design

To access and display the underlying data in a comfortable manner, it is certainly necessary to find an adequate display mode. A special emphasis is put on the development of resources that follow well considered design principles. The main task in developing these applications was to devise a design and a web interface such that access to the text and related information could be fully utilized.

1.10. AAC-Austrian Academy Corpus

Having established a working infrastructure for the digital texts available, the AAC is developing more sophisticated methods of utilizing large scale corpora on the basis of various systems, whereby it is regarded as crucial to provide valuable historical text sources based upon the principles of corpus research for research in the field of historical text studies and linguistics. The purpose of this paper is to present the basic considerations and research perspectives behind the systematic central access structure for the corpus holdings provided by the "AAC-Austrian Academy Corpus" at the "Institute for Corpus Linguistics and Text Technology", the AAC Container.

2. References

AAC-Austrian Academy Corpus: AAC-FACKEL. Online Version: "Die Fackel. Herausgeber: Karl Kraus, Wien 1899-1936". AAC Digital Edition No 1 (Editors-in-chief: Hanno Biber, Evelyn Breiteneder, Heinrich Kabas, Karlheinz Mörth), http://www.aac.ac.at/fackel

AAC-Austrian Academy Corpus and Brenner-Archiv: BRENNER ONLINE. Online Version: "Der Brenner. Herausgeber: Ludwig Ficker, Innsbruck 1910-1954", AAC Digital Edition No 2, (Editors-in-chief: Hanno Biber, Evelyn Breiteneder, Heinrich Kabas, Karlheinz Mörth), http://www.aac.ac.at/brenner

Mörth, Karlheinz (2002): The representation of literary texts by means of XML: some experiences of doing mark-up in historical magazines. In: Michael Fraser, Nigel Williamson and Marilyn Deegan (Eds.), *Digital Evidence. 2002. Selected papers from DRH 2000, Digital Resources for the Humanities Conference*. Office for Humanities Communication 14, pp. 17-32.

Efficient N-gram Language Modeling for Billion Word Web-Corpora

Lars Bungum and **Björn Gambäck**

Norwegian University of Science and Technology
Trondheim, Norway
{larsbun,gamback}@idi.ntnu.no

Abstract

Building higher-order n-gram models over 10s of GB of data poses challenges in terms of speed and memory; parallelization and processing efficiency are necessary prerequisites to build the models in feasible time. The paper describes the methodology developed to carry out this task on web-induced corpora within a project aiming to develop a Hybrid MT system. Using this parallel processing methodology, a 5-gram LM with Kneser-Ney smoothing for a 3Bn word corpus can be built in half a day. About half of that time is spent in the parallelized part of the process. For a serial execution of the script, this time usage would have had to have been multiplied by 250 (corresponding to close to two months of work).

1. Introduction

As part of the language modeling in, for example, a statistical machine translation systems, it is necessary to build n-gram models over very large corpora. The purpose of the language modeling is to help the machine translation system select the correct translation candidate of many, as the graph of possible translations is searched. It is possible to include the language modeling while searching the graph, as well as invoking a separate disambiguation module to select between candidates for a particularly difficult word.

Building higher-order n-gram models over 10s of GB of data poses challenges in terms of speed and memory; parallelization and processing efficiency are necessary prerequisites to build the models in feasible time. The paper describes the methodology developed to carry out this task on web-induced corpora within PRESEMT ("Pattern REcognition-based Statistically Enhanced MT"; http://www.presemt.eu), a project developing a hybrid statistical Machine Translation (MT) system. The establishment of a framework which allows for the rapid creation of new large language models of high order is a necessity in such an application.

The n-gram models were built with the standard tool IRSTLM, the IRST Language Modeling Toolkit (Federico and Cettolo, 2007). The IRSTLM framework was adapted to the OpenPBS queue handler in order to distribute the task to a cluster of machines.

The alternative to adapting the parallelization scripts from IRSTLM — or the similar SRILM (Stolcke et al., 2011), or some other already existing, openly-available language modeling toolkit, e.g., RandLM or KenLM described below — would have been to implement a n-gram modeling tool in a parallel programming framework such as MPI or OpenMP. This approach was discarded because it was unlikely to result in any significant gain in performance over the chosen PBS alternative, which produced acceptable results.

Using the adapted scripts we were able to provide the PRESEMT project with higher-order (5- and 7-gram) language models, built and rebuilt according to the needs of the project. Language models of various sorts (lemma-based, word-based, POS-based or combinations thereof) were built, to provide extended information to the search algorithms.

The rest of the paper is laid out as follows. The next section discusses some previous attempts to build very large scale language n-gram models, in particular in the context of statistical machine translation. The corpora used as basis for creating the n-gram models in the present work are described in Section 3. Thereafter the experimental setup and methodology is detailed in Section 4. The core of the paper are the experimental results and statistics on the created models which are given in Section 5. The final section then sums up the discussion and points to the conclusions that can be drawn.

2. Related Work

Brants et al. (2007) investigate how large language models can be built using distributed techniques. They report decreasing perplexity and better n-gram coverage as the number of tokens increase, and also show how the larger n-gram models improve BLEU (Papineni et al., 2002) scores for a given MT task. The approach taken for the distributed compilation of language models is the Map-Reduce framework, where the counting step of the n-gram model creation is parallelized. This process is separated into a *mapping* step where words and keys are gathered, after which they are *reduced* into separate processes making sure the counting of the same keys are done on the same machines. For a 30G size corpus, Brants et al. report a computation time of two days for language models built with Kneser-Ney smoothing.

Several avenues have been taken to the problems of storing and processing huge n-gram language models. To this end, Talbot and Osborne (2007) use a Bloom filter with logarithmically quantized n-gram frequency counts, that is, a *lossy* randomized representation efficiently encoding the n-grams together with their frequency information. This Randomised Language Modelling (now commonly referred to as 'RandLM') can give significant storage space reductions but at the cost of some extra false positives and reduced decoding speed.

In contrast, Pauls and Klein (2011) discuss a number of compact *lossless* implementations based on tabular tries storing only the suffix of the n-gram (the last word) together with an offset encoding the context (the remaining words). Working on the 4B n-gram Google corpus, they encode each n-gram in only 23 bits, in the best case reducing storage requirements to only 1/4 and improving even on the best previous lossy representations. Encoding the context also gives faster processing (since there is no need to look up the context again when moving on to the next word). Combining this with using a direct-mapped cache, Pauls and Klein report obtaining substantial speed-ups (up to 300%).

In a similar fashion, Heafield (2011) introduces a language modeling library called 'KenLM'. He compares regular hash tables to using tries and shows that a linear probing hash table method gives significantly faster processing while the tries produce a lot smaller data structures. Heafield also discusses a lossy compression of the trie pointers which further reduces the necessary storage space, but concludes that the linear probing hash tables are preferable if processing speed is more important than reduced memory usage. On the other hand, RandLM is potentially the currently most memory efficient approach, even though the memory allocation needed by the tries can be further optimized by lossless compression, as shown by Raj and Whittaker (2003).

3. Corpora

In the development of the language modeling scripts, three corpora of English ('enTenTen', with 3.5Bn Words), German ('deTenTen', 3.2Bn Words), and Italian ('itTenTen', 2.2Bn Words) were used, all originating in the previous "Web as Corpus" corpora known as, respectively, UKWaC, DeWaC and ItWaC (Baroni and Kilgarriff, 2006).

The corpora were mined from the web and processed into a "vertical" corpus format, with one word occupying one line, displayed in various forms: the original form, lemma, part-of-speech (POS), or the combinations thereof (Kilgarriff et al., 2010; Kilgarriff et al., 2011).

3.1. Noise in the corpora

After the tagging, the corpora still contained noise, that had to be addressed before building the models. Many of the problems stem from the web-corpora being encoded in a mixture of character sets.

For example, the most notable sources of noise in the German corpus were:

1. Words beginning with special characters (*-Bus*).
2. Higher order special UTF characters (different newlines and so on). This created some headaches before a proper solution could be found. With `less`, the characters get rendered like `U+0084`, etc., but with `cat` and `more` they are invisible (as they perhaps should be). In the LM software they turn up as "token ghosts".
3. Umlauts being rendered differently (from various character set).
4. Words that are split up that clearly are supposed to be one word (such as "*Bewaff- net*").
5. Repeated strings to three occurrences (the corpus introduced many tokens of repeated words).
6. Very long words (usually 50 or 100 characters, possibly created by hammering the keyboard) .

Scripts were written to mitigate these effects, as well as unifying different representations of dates and numbers into the collection tokens `@date` and `@card`.

3.2. Preprocessing

All the corpora were tokenized and part-of-speech tagged with the TreeTagger (Schmid, 1994), and presented in a "vertical format", where each word used one line, and the different forms of the word — original form, lemma, POS, and special lemma + POS (called lempos) — were printed, in a tab-separated form.

Using the data more or less in the same form as it was retrieved from the web was a project research goal in its own right. However, it proved necessary to do some preprocessing to get workable data out of the corpora. Most notably stripping higher-order UTF characters that would crash the IRSTLM or give undesired output, for example, "words" rendered as spaces which would produce spurious n-grams. Hence, before building the language models the corpora were transformed from the vertical format to a horizontal format with one sentence per line encapsulated in `<s> </s>` sentence boundary markers.

Especially the German corpus produced a very large number of unique tokens. The highly compounding nature of the German language resulted in many tokens ending in a "–", as the compounds were used in split mode (enumeration or linebreak) in the web material. Date formats also varied a lot, giving rise to many spuriously unique tokens.

4. Methodology

A 96 node cluster partitioned in equal parts of nodes with 48G and 24G RAM was used to perform the experiments. The cluster uses the Linux operating system, and the OpenPBS job scheduler (`http://www.mcs.anl.gov/research/projects/openpbs`). The IRSTLM software package already had scripts for parallel treatment of data developed for another (closed) version of the PBS system, and this was changed to adhere to the slightly different syntax of OpenPBS.

The processing cycle goes through the following steps:

1. A dictionary is compiled for the whole input corpus.
2. The corpus is sectioned into n sections according to word frequency.
3. N-grams are counted for each of these sections.
4. (Sub-)LM scores are computed for the sections.
5. Files are merged into one LM.

The sectioning of the dictionaries (or the unigram) lists balanced with regard to frequency, would create a list of unigrams for each section, totaling to a similar sum. To reach this sum you would need more unigrams as the frequencies go down. The top unigrams would alone constitute a frequency well above the average frequency per section, but a list with only one entry can not be split up. Steps 3 and 4 are the steps that are carried out in parallel on each node. A bash script submits the jobs to the PBS queue and tells the jobs to delay merging until all jobs have successfully finished.

Due to resource constraints, IRSTLM uses similar scripts to section up the building of the LMs also when running on a serial architecture, as memory requirements of building a large model could easily exhaust any machine. Instead of carrying out the steps above after another, the tasks could be submitted to a queue handler, performing them in parallel (with the exception of the first step, dictionary collection, and the last step, merging).

Since the same scripts were used, the parallel processing reliably gave the same output as serial processing, and the speed-up factor (though influenced by the load on the cluster) was easy to assess.

The changes in adapting the Sun Grid PBS script to the OpenPBS format mostly related to the control of work flow. It is possible to specify that specific jobs submitted to the queue will be halted until the successful execution of others. To avoid submitting too many jobs at once, the shell script waited for the dictionary compilation before the n-gram counting and sub-LM (as described above) jobs were submitted. The jobs were sent to the queue at the same time, where each individual sub-LM job depended on the corresponding job for n-gram counting, as the n-gram counts for each section needed to be compiled before their respective sub-LM (LM for that section) could be derived. This was controlled by letting each sub-LM job submitted to the queue handler depend on the successful execution of the corresponding n-gram counting job.

The IRSTLM framework can output LMs in an internal format, the ARPA LM format, as well as a compiled version for quicker access with IRSTLM tools (the local platform is Linux/amd64, but the compile step can be done on any architecture).

5. Results

The corpora mentioned in Section 3. were sectioned into `TRAIN` and `TEST` corpora with a Perl script randomly drawing 10% of the lines into the latter for testing purposes.

The sizes of the corpora are shown in Table 1.

For each language, two corpora were extracted, a lemma corpus, containing only the lemmata in succession and one corpus consisting of full forms. The minor differences in size are explained by idiosyncrasies in the extraction methods from the original format of the corpora.

For all corpora, 5- and 7-gram models were built with Kneser-Ney smoothing. For the *fullform* corpora, 5-gram models with and without pruning of singleton n-grams (i.e., n-grams occurring only once in the corpus) were built as well. Due to preprocessing discarding some tokens from the *fullform* corpora, that were not discarded in their lemmatized versions, the number of words reported in Table 1 are lower for the *fullform* corpora.

English	`TRAIN`		`TEST`	
	Lines	Words	Lines	Words
Lemma	108.4	3150.5	12.0	350.3
Fullform	107.4	3122.6	11.9	347.0

(a) Size of the English Corpora

German	`TRAIN`		`TEST`	
	Lines	Words	Lines	Words
Lemma	141.8	2837.0	15.8	315.3
Fullform	141.1	2809.0	15.7	312.2

(b) Size of the German Corpora

Italian	`TRAIN`		`TEST`	
	Lines	Words	Lines	Words
Lemma	78.5	2913.4	8.7	323.9
Fullform	77.9	2851.2	8.7	317.0

(c) Size of the Italian Corpora

Table 1: Size of training and test corpora.
Figures reported in million lines and words.

5.1. Corpus Statistics

Evaluating language models is most interesting relative to a specific task such as Machine Translation (MT) or spell checking. In this work, no specific task was employed, and the held-out test corpora were instead used to compute perplexity and out-of-vocabulary (OOV) statistics. For the pruned models, perplexity statistics on the `TEST` corpora were collected. For the unpruned corpora this was not possible due to memory constraints.

The IRSTLM (Federico et al., 2010) language modeling software's standard functionality offers computation of the above-mentioned statistics for a corpus. This was utilized for all three languages and all corpus types.

The results on the `TEST` corpora for nine language models are shown in Table 2. The difference in word numbers in the `TEST` between the dictionary sizes shown in Table 1 are explained by how the sentence boundary markers are counted. In the former table, they are counted once per sentence, whereas all markers are counted in the latter, making the difference equate to the line number.

For the Lemma-based language models, the 7-gram models have the lowest perplexity, whereas the opposite is the case for the Fullform-based models where the 5-gram models have lower perplexity (although just barely) for English and Italian, with and without the effect of the OOV words taken into account. An exception to this picture is the German corpus, where the 7-gram model has markedly lower perplexity with the 7-gram model also for the Fullform version of the corpus.

The difference in perplexity between the Lemma and Fullform-based models are markedly greater for the DE

English	Nw	PP	PPwp	Noov	OOV
5-Lemma	338,4	66,466.56	1,483.36	517,501	0.15%
7-Lemma	338,4	65,720.84	**1,466.72**	517,501	0.15%
5-Fullform	335,1	59,605.54	**1,432.67**	564,070	0.17%
7-Fullform	335,1	59,812.53	1,437.65	564,070	0.17%

(a) EN TEST Corpus

German	Nw	PP	PPwp	Noov	OOV
5-Lemma	299,6	29,740.41	2,088.30	13,46,978	0.45%
7-Lemma	299,6	29,264.45	**2,054.88**	13,46,978	0.45%
5-Fullform	296,5	62,139.56	4,606.35	1,422,892	0.48%
7-Fullform	296,5	60,932.71	**4,516.89**	1,422,892	0.48%

(b) DE TEST Corpus

Italian	Nw	PP	PPwp	Noov	OOV
5-Lemma	315,2	81,825.78	1,682.07	433,929	0.14%
7-Lemma	315,2	79,060.37	**1,625.22**	433,929	0.14%
5-Fullform	308,3	96,281.84	**2,208.01**	482,946	0.16%
7-Fullform	308,3	99,458.00	2,280.85	482,946	0.16%

(c) IT TEST Corpus

Table 2: Statistics of the TEST corpora.

N_w is the total number of words in the evaluation corpus,
PP is the perplexity, and
PP_{wp} reports the contribution of out-of-vocabulary (OOV) words to the perplexity.
The out-of-vocabulary word term OOV is defined as $N_{oov}/N_w * 100$,
with N_{oov} being the number of OOV words.

corpus, suggesting a higher degree of noise in this corpus as discussed in Section 3.1.

5.2. Dictionary Growth Curves

In addition to the statistics above, a *dictionary growth curve* was obtained, that is, a curve showing the amount of n-grams above the orders 0–9, with the OOV frequency in each category when testing on the TEST corpus.

The Dictionary Growth Curves (DCGs) are shown in Tables 3a–3f. The first three columns of each of the tables show the percentage of words in the training corpus whose frequencies are over 0 (all of them, 100%), those having a frequency over 1 (40%), etc.

The reader will notice that the number of dictionary entries (unigrams) should be the same as the number of unigrams in an unpruned language model. However, due to the implementation Kneser-Ney smoothing, the singleton n-grams for order $1, 2$ were pruned away in the process.

Again, the markedly higher number of dictionary entries for the DE corpora stand out.

5.3. N-grams Counts

We also extracted the numbers of each n-gram level from the corpora, which easily done by looking at the header of the output language model files.

The number of n-grams in the models for all three languages are shown in Table 4. For the English and Italian corpora, the 4-grams were the most frequent n-gram order in the language models, whereas the highest amount of non-singleton n-grams was found in the bigram category for the German corpus.

In addition, unpruned models were built for the Fullform corpora. Comparing those to the singleton-pruned language models shows that the amount of n-grams quickly gets enormous. When zipped down, the unpruned models required about 31G of storage (in the intermediate iArpa format), and were thus not very workable for standard machinery. Using the quantization functionality of the IRSTL toolkit might be a solution to this.

5.4. Computation Times

Because of the differences in load on the cluster over time, it is difficult to report accurate computation times. However, as an indication, the whole process of building any one of these language models would take 8–12 hours depending on the cluster load. If the load was low, it was possible to build three such models simultaneously within the same time frame.

Freq	Entries	Percent	Freq	OOV
>0	7,507,448	100.00%	<1	0.15%
>1	3,072,234	40.92%	<2	0.22%
>2	2,074,727	27.64%	<3	0.26%
>3	1,628,826	21.70%	<4	0.29%
>4	1,362,639	18.15%	<5	0.32%
>5	1,185,035	15.78%	<6	0.35%
>6	1,054,988	14.05%	<7	0.37%
>7	956,259	12.74%	<8	0.39%
>8	877,378	11.69%	<9	0.41%
>9	813,647	10.84%	<10	0.43%

(a) DCG for English Lemma Corpus

Freq	Entries	Percent	Freq	OOV
>0	8,203,706	100.00%	<1	0.17%
>1	3,335,431	40.66%	<2	0.24%
>2	2,280,889	27.80%	<3	0.28%
>3	1,801,269	21.96%	<4	0.32%
>4	1,516,574	18.49%	<5	0.35%
>5	1,325,480	16.16%	<6	0.38%
>6	1,185,591	14.45%	<7	0.40%
>7	1,079,341	13.16%	<8	0.43%
>8	994,459	12.12%	<9	0.45%
>9	925,466	11.28%	<10	0.47%

(b) DCG for English Fullform Corpus

Freq	Entries	Percent	Freq	OOV
>0	19,300,334	100.00%	<1	0.45%
>1	7,404,928	38.37%	<2	0.64%
>2	4,841,598	25.09%	<3	0.76%
>3	3,706,439	19.20%	<4	0.86%
>4	3,042,511	15.76%	<5	0.94%
>5	2,606,399	13.50%	<6	1.01%
>6	2,293,273	11.88%	<7	1.07%
>7	2,056,786	10.66%	<8	1.13%
>8	1,870,568	9.69%	<9	1.18%
>9	1,719,753	8.91%	<10	1.22%

(c) DCG for German Lemma Corpus

Freq	Entries	Percent	Freq	OOV
>0	20,775,474	100.00%	<1	0.48%
>1	8,163,441	39.29%	<2	0.68%
>2	5,463,441	26.30%	<3	0.81%
>3	4,253,230	20.47%	<4	0.92%
>4	3,543,995	17.06%	<5	1.01%
>5	3,071,589	14.78%	<6	1.09%
>6	2,731,179	13.15%	<7	1.15%
>7	2,470,988	11.89%	<8	1.21%
>8	2,263,809	10.90%	<9	1.27%
>9	2,096,181	10.09%	<10	1.32%

(d) DCG for German Fullform Corpus

Freq	Entries	Percent	Freq	OOV
>0	6,475,359	100.00%	<1	0.14%
>1	2,778,546	42.91%	<2	0.20%
>2	1,903,603	29.40%	<3	0.24%
>3	1,511,911	23.35%	<4	0.27%
>4	1,275,415	19.70%	<5	0.30%
>5	1,116,931	17.25%	<6	0.32%
>6	1,000,423	15.45%	<7	0.34%
>7	911,485	14.08%	<8	0.36%
>8	840,714	12.98%	<9	0.38%
>9	782,787	12.09%	<10	0.40%

(e) DCG for Italian Lemma Corpus

Freq	Entries	Percent	Freq	OOV
>0	7,365,655	100.00%	<1	0.16%
>1	3,169,535	43.03%	<2	0.22%
>2	2,222,344	30.17%	<3	0.27%
>3	1,786,668	24.26%	<4	0.31%
>4	1,525,238	20.71%	<5	0.34%
>5	1,348,829	18.31%	<6	0.37%
>6	1,219,203	16.55%	<7	0.39%
>7	1,119,484	15.20%	<8	0.41%
>8	1,040,111	14.12%	<9	0.44%
>9	974,612	13.23%	<10	0.46%

(f) DCG for Italian Fullform Corpus

Table 3: DCG curves for the six corpus types

On average, the parallelized jobs would take about 1.5 hours, with the exception of the jobs counting n-grams based on the most frequent unigrams, that could take up to 5 hours to finish. The merging of the sub-language models would have to wait for all the sub-models to finish, and hence it was necessary to wait for these initial jobs to exit. Theoretically it would be possible to find an ideal number of jobs to minimize the total computation time, where the smaller jobs would be made big enough to correspond to the jobs based counting n-grams for the most frequent unigrams. This would lead to a smaller total computation time, being easier on the cluster, but would not change the total time the script

English	Fullform		
Order	5 pruned	5 unpruned	7 pruned
1-gr	3.3	3.3	3.3
2-gr	135.3	135.3	135.2
3-gr	165.6	668.8	165.6
4-gr	**222**	1,451.9	**222.0**
5-gr	179.4	**2,026.2**	179.4
6-gr			115,4
7-gr			72,1
In total	705.7	4,285.5	893.3

(a) EN Corpus n-gram counts

German	Fullform		
Order	5 pruned	5 unpruned	7 pruned
1-gr	8.1	8.1	8.1
2-gr	237.0	237.0	**237.9**
3-gr	168.5	842.9	168.5
4-gr	**180.5**	1,493.3	180.5
5-gr	128.1	**1,844.2**	128.1
6-gr			79.6
7-gr			52.6
In total	722.4	4,425.5	854.7

(b) DE Corpus n-gram counts

Italian	Fullform		
Order	5 pruned	5 unpruned	7 pruned
1-gr	3.1	3.1	3.1
2-gr	131.2	131.2	131.2
3-gr	169.7	670.3	169.7
4-gr	**225.1**	1,454.8	**225.1**
5-gr	174.8	**1,991.3**	174.8
6-gr			114.2
7-gr			79.2
In total	704.1	4,250.9	897.6

(c) IT Corpus n-gram counts

Table 4: N-gram counts for pruned and unpruned 5,7-gram models from the Fullform EN/DE/IT corpora. Figures reported in million n-grams.

needs to return with a language model.

6. Conclusions

Experimenting with building and rebuilding n-gram models built from large corpora requires efficient computation. In this paper we have shown how they can be efficiently built using the IRSTLM framework, adapted to the OpenPBS job scheduler. Although the machinery used can be considered high-end, such equipment is available for many universities and research organizations today.

With web corpora. noise can be a problem, and we have identified steps that can be taken to reduce the number of unique tokens that are not members of the language, but rather have been produced as the result of idiosyncrasies in the corpus processing.

Comparing models of different orders and built on lemmas, nouns, verbs, etc., in a final application (in this case Machine Translation) is also of value. When dealing with large corpora, it is possible to extract valuable linguistic information about the languages, as the perplexity of corpus samples of a language wanders asymptotically towards the perplexity of the language itself.

It is interesting to note the low degree of out-of-vocabulary words, also when using corpora that retain capitalization and inflected forms (as in the Fullform corpora above), an indication of the benefit of large data.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement nr 248307 (PRESEMT). Thanks to André Lynum, Erwin Marsi and the anonymous reviewers for useful comments on earlier drafts of the article.

7. References

Marco Baroni and Adam Kilgarriff. 2006. Large linguistically-processed web corpora for multiple languages. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pages 87–90, Trento, Italy. ACL.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic, June. ACL.

Marcello Federico and Mauro Cettolo. 2007. Efficient handling of n-gram language models for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 88–95, Prague, Czech Republic, June. ACL.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2010. *IRST Language Modeling Toolkit, Version 5.50.02: User Manual*. FBK-irst, Trento, Italy, November.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. ACL.

Adam Kilgarriff, Siva Reddy, Jan Pomikálek, and Avinesh PVS. 2010. A corpus factory for many languages. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 904–910, Valletta, Malta, may. European Language Resources Association (ELRA).

Adam Kilgarriff, Avinesh PVS, and Jan Pomikálek. 2011. Comparable corpora BootCaT. In Iztok Kosem and Karmen Kosem, editors, *Proceedings of eLex 2011, Electronic Lexicography in the 21st Century: New Applications for New Users*, pages 122–128, Bled Slovenia, November. Trojina, Institute for Applied Slovene Studies.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. ACL.

Adam Pauls and Dan Klein. 2011. Faster and smaller n-gram language models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 258–267, Portland, Oregon, USA, June. ACL.

Bhiksha Raj and Ed Whittaker. 2003. Lossless compression of language model structure and word identifiers. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, page 388–391. IEEE.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, volume 12, pages 44–49, Manchester, UK.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at Sixteen: Update and outlook. `http://www-speech.sri.com/pubs/papers/Stol1112:SRILM/document.pdf`.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 512–519, Prague, Czech Republic, June. ACL.

Large Mailing List Corpora: Management, Annotation and Repository

Damir Cavar, Helen Aristar-Dry, Anthony Aristar

Institute of Language Information and Technology, Eastern Michigan University
2000 Huron River Drive, Suite 104, Ypsilanti MI 48197
E-mail: {damir,hdry,aristar}@linguistlist.org

Abstract

Corpora which grow continuously require different maintenance procedures from static text collections. In the following, we present an infrastructure for corpus maintenance, linguistic analysis and annotation, and storage and retrieval infrastructure that is used for dynamic and continuously growing corpora like the LINGUIST List Mailing List corpus and other such restricted domain professional mailing lists. We describe an architecture that is based on flexible text storage, annotation and retrieval using text and language processing tools for automatic meta- and linguistic annotation, also using common linguistic analysis components as part of GATE or UIMA, and current storage systems that are more flexible and dynamic than common relational databases, as for example XML-databases and NoSQL storages. This architecture and environment not only serves our current needs for advanced annotation and search over the mailing list archives; it also is the foundation of a future corpus environment and service.

1. Introduction

The LINGUIST List (LL) is hosting numerous mailing lists that are related to linguistic topics. It provides many different linguistic tools and services, among which are online accessible corpora, infrastructure and related corpus analysis tools. Currently the mailing list corpora are stored in the original Listserv mail format, as well as in database tables using a relational database server. HTML versions of the mails are generated dynamically from the two source formats for online access from the LINGUIST List pages. In an attempt to redesign a more advanced search functionality, we created an infrastructure to advance search and content analysis using automatic annotation for the mailing list corpus, we developed conversion, annotation and storage concepts that facilitate new ways to create, annotate, and work with text corpora. The initial infrastructure serves several purposes and opens up new possibilities to deal with dynamic text corpora and individual user needs for corpus analysis and handling. More precisely, we are focusing on solutions related to issues of an ideal technical backend, as well as a future user front-end that includes new concepts for corpus tools and work:

- Dynamic growth of content of corpora: for example, when considering only the mailing lists archived at LINGUIST List, these need to be annotated with meta-information and with linguistic information, and made available for search instantly with minimized computation time and technical overhead.
- Virtualization of corpora: users should be able to define sub-corpora for search and analysis, annotate and upload their own corpora in a future interface, and define individual qualitative and quantitative analyses.
- Dynamic annotation: users should be able to annotate existing corpora using their own automatic tools, or to manually annotate the data with their own annotation standards.
- The annotation should be mapped on an annotation interlingua or standard such as the General Ontology for Linguistic Description (GOLD) for interoperability reasons.

The availability of technologies for storage, annotation, linguistic processing and analysis, even machine translation, makes new approaches to corpora possible. So do new quantitative analysis methods, environments and tools. The integration of these technologies into a centralized corpus infrastructure was one of our goals, with the potential to provide the community with tools and services related to corpora. The initial architecture and infrastructure, described in the following paragraphs allows us to evaluate the scalability of the tools along the given parameters: dynamic growth of large-scale corpora, and instant availability of search indexes and annotations (meta and linguistic).

2. The LINGUIST List Corpus

As an initial development and evaluation corpus we use the LINGUIST List (LL) mailings, henceforth LINGUIST List Corpus (LLC). This mailing list corpus consists of currently 61,626 mail submissions to LL. Being an actively used list, the number of submissions is growing continuously. This is a moderately dynamic corpus that allows us to test an automatic annotation pipeline and search interface. Currently these mail submissions are available in the following data formats:

- a relational database (SQL-based)
- an archived listserv format
- generated HTML pages for the web interface

In addition to the LL mails, LL also stores 238 mailing lists in the Listserv text format that currently have a space requirement of 5.6 GB on disk. These mailing lists are dynamically growing too. Currently all mailing lists require 6.1 GB together with the LL mailings archive. The LL mailings are also stored in a relational database. There is a difference between the database storage of submissions and the real message sizes due to templating in the database. In fact, for the LL mailings it is the case that they are initially submitted to the database, edited and moderated, and then only mails are generated and send to the subscribers. Some of the submissions are input to the system using structured web interface pages, thus the content is partially typed and denotes person, location, and institution names, or time expressions and roles of people and institutions, or their specific type (e.g. educational institution, publisher, author, editor, thesis supervisor). This initial typing of strings is available in the database directly, and thus some of the named entities in the mailings can be annotated quite easily. More information can be added using the advanced semantic role properties or types of mentioned institutions and person names.

As mentioned, the Listserv archive of all the available mailing lists is continuously growing, with many highly active lists like the LL mailing list, Corpus and Childes lists. Currently we estimate there are 195,782 postings in the Listserv archive alone.

Our specific interest in these mailing lists is not only driven by the fact that we want to archive them and make them searchable. We are also highly interested in their content, and want to make the language and linguistics-specific professional knowledge and information available for mining and further analysis.

The challenge with handling this type of corpus is its continuous growth and the different file formats and data structures, that make a unique annotation, and search and analysis difficult to achieve.

3. The Corpus Management Process

From the existing file formats of the corpora (e.g. Listserv mail, DB tables and fields, generated HTML), we created different intermediate and final export formats. In order to make an advanced search and analysis of the mailing lists possible, the content has to be augmented with semantically typed meta-information, as well as linguistic annotation. As mentioned, most of this information is available as semi-structured information in raw mail file headers in the Listserv format, or in the semantics of the database tables and fields. The obvious and common format for making these pieces of information explicit for further computation is XML. Consider the following sample:

```
<dissertation id="2749">
  <disstitle>some title</disstitle>
  <institution_name>University of Edinburgh
  </institution_name>
  <progtitle>School of Informatics
  </progtitle>
  <degreedate>2009</degreedate>
  <dissstatus>In Progress</dissstatus>
  <dissabstract>...</dissabstract>
  <people>
    <person id="111660" role="Author">
      <personfn>David</personfn>
      <personmi></personmi>
      <personln>Smith</personln>
      <institution>Universitt Potsdam
      </institution>
    </person>
    <person id="653"
      role="Dissertation Director">
      <personfn>John</personfn>
      <personmi></personmi>
      <personln>Johnston</personln>
      <institution>University of Edinburgh
      </institution>
    </person>
  </people>
```

It shows the specific fields and data of a dissertation abstract mailed on LL. The `<dissabstract>` field has been reduced and contains the text of the dissertation abstract. For any specific Diss-mail the number of involved people and institutions can vary. The database source of the LL mailing list makes extracting these specific annotations easy. For the other mailing lists, specific parsers and text processing tools are necessary to extract and annotate such information.

We established the initial conversion process with the aim of enabling not only advanced corpus analyses over the resulting format, including concordances, keyword-in-context views and collocation analyses, but also scalable and adaptable linguistic annotation either by exporting the database fields and relations of the LL mailings to an appropriate XML-structure, or by processing with specific scripts the raw text email submissions in other mailing lists. We decided to use the TEI P5 XML (Burnard and Bauman, 2007) format as the storage and processing format. The following graph shows the initial conversion and annotation pipeline set up for the LLC:

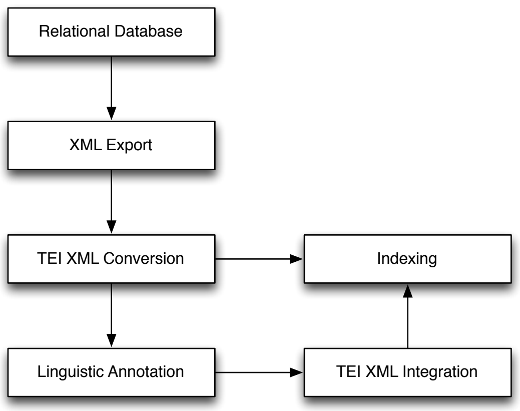


Figure 1: The corpus generation process

Since the complete LINGUIST List mailing list corpus is stored in a relational DB, we make it available in a XML format that is using the DB storage structure and corresponding field names and tables. In a subsequent conversion step these initial XML-export formats are converted to a standardized TEI P5 XML format using XSLT and specific conversion scripts.

The XML format is enriched with meta-information to enable search, filtering and selection of specific topics used in the LINGUIST List classification system, as well as for example timeline information. Besides the TEI-compatible title statement with the mailing title and editor information, the publication information is added, containing publication date and volume numbers, which enables timeline selection for search restrictions. Of particular interest is meta-information about the linguistic fields and topic, or the languages used and discussed. The following XML-sample shows how for example the TEI-compatible class declarations are integrated into the meta-information by using the database information or extracting the semi-structured information from the mailings:

```
<classDecl>
  <taxonomy xml:id="topic">
    <category>
      <catDesc>Topic</catDesc>
    </category>
  </taxonomy>
  <taxonomy xml:id="lingfield">
    <category>
      <catDesc>Linguistic Field</catDesc>
    </category>
  </taxonomy>
</classDecl>
```

The content of the mailings is generated as a section of paragraphs <p> and line-breaks <lb> from the raw emails in the Listserv archive, or directly from the DB fields. The following sample shows a typical first TEI P5 compatible output format:

```
<text>
 <body>
  <div type="message" n="1">
   <head>Message 1: Syntactic Analysis:
         Sobin</head>
   <div type="header">
   <p>Date: <date>20-Dec-2011</date><lb/>
      From: Kristen Holding
       <email>kholding@wiley.com</email><lb/>
      Subject: Syntactic Analysis: Sobin</p>
   <p>Title: Syntactic Analysis<lb/>
      Subtitle: The Basics<lb/>
      Published: 2011<lb/>
      Publisher: Wiley-Blackwell<lb/>
      <ref target="...">...</ref></p>
   <p>Book URL:
      <ref target="...">...</ref></p>
   <p>Author: Nicholas Sobin<lb/>
      Hardback: ISBN: 9781444338959 ...<lb/>
      Paperback: ISBN: 9781444335071 ...</p>
  </div>
  <div>
   <head>Abstract:</head>
   <p>...</p>
  </div>
  <div type="footer">
   <p>Linguistic Field(s):
      Applied Linguistics<lb/>
      Syntax</p>
   <p>Written In: English (eng)</p>
   <p>See this book announcement ...<lb/>
   <ref target="...">...</ref></p>
  </div>
 </div>
 </body>
</text>
```

A sample TEI-XML encoded mailing that results from the initial conversion step of the exported DB-XM-data can be found at the following URL: http://ltl.emich.edu/llc/23-33-TEI.xml.

In a second step the TEI XML files are automatically processes and linguistically annotated and enriched. The different mailing list topics can contain text in different languages, including for example English, German, French, Spanish. We use own linguistic components for the pre-processing and language identification, as well as freely available NLP components like the Stanford CoreNLP tools (http://nlp.stanford.edu/software/corenlp.shtml) or other components contained and distributed in the GATE and UIMA frameworks. The content of the <text>-section of the initial TEI documents is sent to linguistic components. The resulting linguistic annotation is wrapped into TEI P5 XML format and integrated into the TEI P5 file of the mailing. Annotation to TEI XML wrappers are available online, however they do not use a complete TEI P5 format for their output. Our implementation of these wrappers, however, keeps to the TEI P5 standard. It provides a complete mapping of the output formats of the linguistic components that we use, e.g. sentence recognizers, tokenizers, lemmatizers, part-of-speech taggers, named entity recognizers and syntactic parsers to a TEI P5 compatible XML otuput. A simple demo of such a mapping can be found at the following URL http://ltl.emich.edu/txt2tei/.

The initial output of the linguistic annotation step includes the segmentation into sentences, tokenization and annotation of tokens with respect to lemma and part-of-speech information, as well as named entity augmentation. A sample linguistic annotation of a paragraph in the <text>-section looks as follows:

```
<p>
 <w lemma="title" type="NN">Title</w>
 <pc>:</pc>
 <w lemma="syntactic"
  type="JJ">Syntactic</w>
 <w lemma="analysis"
  type="NN">Analysis</w><lb/>
 <w lemma="subtitle"
  type="NN">Subtitle</w>
 <pc>:</pc>
 <w lemma="the" type="DT">The</w>
 <w lemma="basic"
  type="NN">Basics</w><lb/>
 <w lemma="publish"
  type="VBN">Published</w>
 <pc>:</pc>
 <date>2011</date><lb/>
 <w lemma="publisher"
  type="NN">Publisher</w>
 <pc>:</pc>
 <name type="publisher">
 <w lemma="Wiley-Blackwell"
  type="NNP">Wiley-Blackwell</w>
 </name>
 <lb/>
 <ref target="http://www.wiley.com">
  http://www.wiley.com</ref>
</p>
```

A sample corpus file can be accessed at the fol-

lowing URL http://ltl.emich.edu/llc/23-33-TEI-Ling.xml.
Currently the setting we are using uses the plain NLP-components and specific components for TEI XML file formats. For evaluation purposes, we are expanding the architecture to include two different text processing and analysis architectures, in particular the General Architecture for Text Engineering (GATE) (The GATE Team, 2011) and the Apache Unstructured Information Management Architecture (UIMA) (Götz and Suhre, 2004), (Ferrucci and Lally, 2004). These architectures are scalable and allow for almost effortless adaptation of the linguistic processing pipeline. Our goal is, on the one hand to establish linguistic annotation pipelines that can be used for our initial corpus, i.e. the LLC, and automatic annotation of every new mail on one of the hosted mailing lists, generating TEI P5 compatible XML files. On the other hand, we aim at providing a web service for TEI P5 compatible automatic linguistic annotation of corpora that covers as many languages and NLP-components as possible.
These resulting corpus files are subsequently processed and converted into data structures necessary for indexing and different corpus interfaces.

4. Corpus Indexing and Interfaces

As an initial corpus frontend we use Philologic 3. PhiloLogic is the primary full-text search, retrieval and analysis tool developed by the ARTFL Project and the Digital Library Development Center (DLDC) at the University of Chicago.[1] It supports the basic TEI XML meta-data and file format and structuring tags, tokenization and lemma annotation, but without modification not search and processing of deep linguistic annotation of for example syntactic structures in the TEI P5 style. One interface to the LLC is available at the following URL:

http://ltl.emich.edu/llc/

The Philologic interface offers basic search functions, search with regular expressions, Concordance and Keyword in Context presentation. It also allows for the inclusion of meta-data in search, and permits restrictions and extended result overviews. It has an acceptably fast interface and backend, one that also provides context and collocation analysis. Philologic is currently under development. Version 4 of Philologic should appear as a completely new port to Python.
However, one of the drawbacks of using systems like Philologic for online corpus analysis is that the processing and re-indexing of the corpus files is computationally demanding and time consuming. With every new file that is added to the corpus, the complete index has to be re-generated, i.e. all the previous corpus texts have to be re-indexed all over again. Such a solution is not feasible for the dynamic corpus that we want to be able to deal with, in particular not for some of the mentioned aspects of virtualization of corpora, dynamic creation and annotation of textual data, and other aspects of qualitative and quantitative analyses.
Among the core goals of the infrastructure we are designing is to allow for dynamic changes of the annotation levels, the used language models and natural language processing tools. The work with corpora should allow for dynamic additions of corpus content and instant qualitative and quantitative analyses. Sub-corpora should be created without complex re-indexing phases and annotation steps.
In the current phase our initial LLC is realized in three different formats, as a DB content distributed over relational tables and fields, as a raw text Listserv archive, and as a TEI P5 encoded and annotated collection of individual mailings. The relational database makes field-based and full text search possible and is efficient even for hundreds of billions of tokens. However, changes and extensions of the database structure and relations are complicated and time consuming: thus extensions for different annotation types and strategies do not seem to be feasible using a relational database-system. To add different linguistic annotation levels or even different tag-sets would require extensive manual effort and increase of the systems complexity. On the other hand, the manipulation and extension of the annotations within the XML-based corpus files is less complicated and can be manipulated automatically. The indexing however can be quite time consuming and redundant, when using common indexing tools and approaches. This is true for systems like Philologic in its current version, and also for alternative indexing approaches, if changes of the corpus base are frequent, or multiple different annotation levels have to be handled.
As an alternative solution to indexing and dynamic data and annotation storage, in addition to the existing DBMS and file-storages, simple and fast NoSQL storages seem attractive. Key-value stores like Redis are ideal to store indexes, on demand merged indexes, N-gram models, meta-data, text and annotations and other secondary data as hash-tables, sets or lists. Rather than storing large indexes and models in DBMS, we evaluate the possibility to store smaller indexes, sets and N-gram models directly from the corpus management tools in NoSQL storages.

5. Conclusion

The approach we have taken in designing a basic infrastructure for corpus creation and management has been motivated by concerns about not just large corpora and extended annotations. We are also taking into account dynamic amounts of text, in particular the continuous growth of a corpus, as well as possibilities for virtualizing corpora and manipulating the search and analysis space for quantitative studies, and extending the annotations and analyses efficiently. The initial architecture of automatic generation of TEI XML encoded corpus files, their linguistic annotation, and subsequent indexing has shown us that the weak points are the annotation quality and the linguistic components, as well as the final indexing and interface components for such architecture.
Our current activities focus on the evaluation of alternative storage concepts for corpora, evaluating XML-data bases like BaseX (Holupirek et al., 2009), or alternative NoSQL

[1] See: http://sites.google.com/site/philologic3/home.

storages (Tiwari, 2011). NoSQL storages or key-value stores like Redis (Macedo and Oliveira, 2011) are particularly promising, since we would like to efficiently extend the corpus content, the annotations, and new analyses, or simply cache intermediate search results (e.g. collocation analyses, N-gram models, statistical results) or crowd sourced annotations. Without the SQL-layer such storages are much more flexible than relational databases, while merging various index files can be handled efficiently in a relational database and Redis. Extensions and additional storage of data structures from extensions in the annotation or quantitative analyses requires no significant additional overhead in Redis, except for the handling code that generates data representations of annotations or quantitative analyses.

6. References

Burnard, L. & Bauman, S. (2007). TEI P5: Guidelines for electronic text encoding and interchange. Text Encoding Initiative.

Ferrucci, D. & Lally, A. (2004). UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4), pp. 327–348.

Götz, T. & Suhre, O. (2004). Design and implementation of the UIMA common analysis system. *IBM Systems Journal*, 43(3), pp. 476–489.

Holupirek, A., Grün, C., Scholl, M.H. (2009). BaseX & DeepFS: Joint storage for filesystem and database. In *Proceedings of the 12th EDBT Conference*, pp. 1108–1111.

Macedo, T. & Oliveira, F. (2011). *Redis Cookbook: Practical Techniques for Fast Data Manipulation*. O'Reilly Media.

The GATE Team. (2011). *Text Processing with GATE (Version 6).* University of Sheffield Department of Computer Science, April.

Tiwari, S. (2011). *Professional NoSQL.* Wiley, September.

Creating and managing large annotated parallel corpora of Indian languages

Ritesh Kumar[1], Shiv Bhusan Kaushik[2], Pinkey Nainwani[1], Girish Nath Jha[2]
[1]Centre for Linguistics, [2]Special Centre for Sanskrit Studies
Jawaharlal Nehru University, New Delhi
E-mail: {riteshkrjnu, shivkaushik.engg, pinkeybhu39, girishjha}@gmail.com

Abstract

This paper presents the challenges in creating and managing large parallel corpora of 12 major Indian languages (which is soon to be extended to 23 languages) as part of a major consortium project funded by the Department of Information Technology (DIT), Govt. of India, and running parallel in 10 different universities of India. In order to efficiently manage the process of creation and dissemination of these huge corpora, the web-based (with a reduced stand-alone version also) annotation tool ILCIANN (Indian Languages Corpora Initiative Annotation Tool) has been developed. It was primarily developed for the POS annotation as well as the management of the corpus annotation by people with differing amount of competence and at locations physically situated far apart. In order to maintain consistency and standards in the creation of the corpora, it was necessary that everyone works on a common platform which was provided by this tool.

Keywords: ILCI, ILCIANN, Annotation Tool, Parallel Corpora

1. Introduction

In recent time, availability of huge annotated corpora has become very essential for the development of language technologies for any language. Since almost all the Indian languages are considered less-resourced languages, it is very necessary to develop extensive language resources for them in order to make them technologically strong and efficient.

This paper presents the challenges in creating and managing large parallel corpora of 12 major Indian languages (which is soon to be extended to 23 languages) as part of a major consortium project funded by the Department of Information Technology (DIT), Govt. of India (Jha, 2009 and Jha, 2010), running parallel in 10 different universities of India (Choudhary & Jha, 2011).

In order to efficiently manage the process of creation and dissemination of these huge corpora, the web-based (with a reduced stand-alone version also) annotation tool ILCIANN (Indian Languages Corpora Initiative Annotation Tool) has been developed. It was primarily developed for the POS annotation as well as the management of the corpus annotation by people with differing amount of competence at locations physically situated far apart. In order to maintain consistency and standards in the creation of the corpora (it is essential for any corpora to be usable in NLP), it was necessary that everyone works on a common platform which was provided by this tool. The use of the tool ensured that the data is saved on a centralized server in a uniform format which could be later utilized for any NLP task. Besides providing administrative and annotation facilities, the tool also provides the facility for creating parallel corpora as well as automatically adapting the linguistic data from any other source. It could also be potentially used for crowd-sourcing the annotation task and creation of language resources for use in NLP.

2. The Corpora

In its first phase, Indian Languages Corpora Initiative (ILCI) was involved in the creation of translated parallel corpora of 12 Indian languages viz., Hindi, Bangla, Oriya, Urdu, Punjabi, Marathi, Gujarati, Konkani, Telugu, Tamil, Malayalam and English. The basic data of the corpora was collected in Hindi (sourced from different written texts like magazines, newspapers, books, etc.) and then it was manually translated into all other languages by the respective language experts. The data was collected from the two domains health and tourism with 25,000 sentences in each domain (with an average length of 16 words per sentence, counting up to around 400,000 words in each domain in each language). The sentences in each language are aligned parallel (along with an alignment up to the word level as far as possible) and each sentence is given a unique ID (details of data collection and the corpora are included in Choudhary and Jha (2011)). In the second phase of the project, 50,000 sentences are being added to the corpus of each language from the two domains agriculture and entertainment.

The corpus of every language has been initially annotated with part of speech information. This work was done manually by each language group. However soon the challenges of annotating such huge corpora by the people physically distributed over different areas began to come to the fore. Two of the major challenges included:

1. It was very difficult to maintain the sanity and uniformity of the data across all the groups since the annotation was being carried out by the people of varying degree of experience and expertise. In such a scenario the annotated data did not carry a uniform structure despite the clear instructions on how to carry out the annotation. Since it was very necessary to maintain the uniformity throughout the corpora so that any meaningful work could be done using these corpora, there was an urgent need to devise a mechanism to ensure this.

2. There also were some very administrative issues that needed some urgent attention. These included keeping track of the progress of every language group and ensuring that the work is completed within the stipulated time period by each member of the consortium.

As a result of these challenges, the idea of a web-based application for managing as well as carrying out the annotation task came to the fore.

3. Managing the Parallel Corpora

Over the last two decades, numerous annotation tools have been created to meet the required demands of various projects. The most popular and well-known annotation tool among them is General Architecture for Text Engineering (GATE) (Cunningham, 2011). However since it is a stand-alone application, it does not provide the facility of managing a physically distributed project. Moreover it also does not have the facility of managing and creating a parallel corpus. Some of the other significant tools include Stuhrenber et al., (2007), Russell et al. (2005), besides numerous others. Bird et al., (2002) came up with a tool which deals with annotations called ATLAS (Flexible and Extensible Architecture for Linguistic Annotations). Kaplan et al. (2010) discussed SLATE (Segment and Link-based Annotation Tool Enhanced) in their paper. It is a web-based annotation tool and addresses 10 annotation needs: (1) managing the role of annotator and administrator, (2) delegation and monitoring work, (3) adaptability to new annotation tasks, (4) adaptability within the current annotation task, (5) diffing and merging (diffing and merging of data from multiple annotators on a single resource to create a gold standard), (6) versioning of corpora, (7) extensibility in terms of layering, (8) extensibility in terms of tools, (9) extensibility in terms of importing/exporting and, (10) support for multiple languages.

However none of these tools are meant to support the requirements of creating and managing translated parallel corpora. Besides the annotation needs mentioned by Kaplan et al. (2010), couple other requirements need to be fulfilled by a tool for creating and managing parallel corpora include -

1) Translation Work: To build parallel annotated corpora, the tool should support the translation of the source data in the respective languages.

2) Quality assurance: It is a key concern as far as full crowd sourcing is concerned but there must be some automated features to check the quality of translated and tagged data.

3) Crowd-sourcing: The tool must be flexible enough to adapt to the needs of crowd-sourcing at any given point of time.

4. The ILCI Annotation Tool

ILCIANN is a server-based web application which could be used for any kind of word-level annotation task in any language. It is developed using Java/JSP as the programming language and is running on Apache Tomcat

4.0 web server. Some of the facilities provided by the tool for managing a large project include the following:

1. User Management and Monitoring Facility: The tool recognises users at three hierarchical levels:

a. Master Admin: The master admin is basically the main administrator of the project, who spearheads the project and overlooks all the language groups working in the project. The major responsibilities of master admin include

i. Uploading the Files: The major responsibility of the master admin is to upload the source files which are to be translated in different target languages. (S)he is also required to upload the translated files in different languages for annotation (if the translation is not done in the tool).

ii. User Management: This step involves creating the login of users who would annotate the data. Only the master administrator has the authority to create/delete/modify the login for the users who are supposed to annotate/translate the data as well as for the individual language administrators. It ensures the safety as well as authenticity of the tagged data, while theoretically giving an opportunity to a huge community to support and help in building language resources for their languages. Further, since the project of creating parallel corpora, by definition, involves multiple languages, therefore, the users and administrators are also assigned to the language on which (s)he is supposed to work. For instance, if x is a Hindi language annotator, (s)he can only work on Hindi data and cannot do any modification (tagging the data, editing the data and saving it) in other language files.

iii. Monitoring the Project: Besides this, the master admin can maintain the time log of the user accounts (which include the details about currently logged in users, login and logout history of different users from any language group), monitor the overall progress of the project (including the amount of work completed), send notices and reminders to the users as well as administrators of individual language groups regarding the progress of the project.

iv. The other functions and facilities of the master admin are the same as for the administrator of individual language groups, discussed below.

b. Administrator (Admin) – For the purpose of management, each language group is assigned to an administrator The following responsibilities are given to an admin in order to facilitate the increased productivity and proper administration of the project:

i. Assigning and Monitoring the Work: The admin could assign a set of maximum 3 files for annotation to a single user at one time (and a new file is assigned only after one of the files is completed). It eliminates any scope for the duplication of effort in a huge project and also ensures that one or more files are not left incomplete. Furthermore it also helps in quality control of the annotation work by ensuring that, in general, only one user works on one file (and even if a file is re-assigned to some other user then a record is maintained). It also helps in keeping a record of the progress as well as the precise achievement of the individual annotators/translators in the project.

ii. Downloading the Files: The files could be downloaded only when each sentence of the file is tagged and only the administrator has the right to download the files.

c. User (Annotator/Translator) – The user is responsible only for the annotation/translation task. They can work on one of the files that have been assigned to them and annotate/translate it. The users, along with the administrators are able to view the progress made in the project in terms of total work that needs to be done and the total work that has been currently completed.

2. Annotation Facility: In its current form, the tool allows the user to annotate the data at the word level. Some of the major features of the annotation facility include:

a. Complete Language and Tagset Independence: There is no restriction at all related to the use of tagset or language and in any given project any tagset could be used for annotation.

b. Limited Intelligence: The tool provides the facility of limited automatic tagging for closed grammatical categories like pronouns, postpositions, conjunctions and quantifiers which reduces the burden of human annotators. The list of words marked for automatic tagging could be modified and edited by the user during the annotation process and the changes made by one user become available to all other users working in the same language in real time.

c. Limited Editing: The users are also allowed to edit the data in case they find some errors (related to the structure, orthography, translation, etc) or they see a mismatch with the source data.

d. Quality Control: In order to ensure that the data is saved properly and the annotation is carried out using only valid tags from the available tagset, the users are presented with an option to choose one of the tags from the tagset and are not given any freedom in assigning the tags (this will prove to be inefficient for very large tagsets and the effort is made to improve it).

3. Translation Facility: The work is under progress to include the facility of translation of the source data into several target languages in the tool. In its initial stage this facility is expected to provide a rough translation with the help of a bilingual dictionary to help the translators and increase their productivity.

4. Adaptation Facility: The tool also has the facility to adapt and modify data from other sources as well as noisy data in such a way that it could be used properly for the annotation work.

5. Conclusions and Future Work

The use of ILCIANN for annotation purposes could help in resolving lots of issues both on the side of the annotators as well as the developers as we could see in the case of the ILCI project. On the one hand it ensures the uniformity of the data without any scope for any noise creeping into it, which becomes inevitable if the annotation work is carried out manually by a large number of annotators. This makes things easier for the developers who want to work with the data. At the same time, since the tool is a web application, a huge number of people could work together in parallel and seamlessly (without actually worrying about what others have completed, since the tool by its very structure eliminates any scope of redundancy) and contribute to the development of the language resources. Thus it could prove to be a very significant tool for creating annotated corpora, especially in smaller and less-resourced languages, with the help of the community and a large number of online contributors.

The tool needs to be further developed to ensure automated checks for quality assurance (always a concern with online crowd sourcing), check the inter-annotator agreement, increase the options for importing/exporting the data in different formats and also include the facility to create corpora from the web automatically.

6. Acknowledgement

The project for creating the parallel corpora as well as the annotation has been funded by the Department of Information Technology (DIT), Govt. of India. We are thankful to them for their co-operation and support. We are very thankful to all the members of the project for their immense contribution to the project . We are also very thankful to the reviewers and the proofreaders of this paper for their invaluable suggestions and improvements.

7. References

Bird, Steven, David Day, John S. Garofolo, John Henderson, Christophe Laprun, and Mark Liberman. (2000). Atlas: *A flexible and extensible architecture for linguistic annotation*. CoRR, cs.CL/0007022.

Choudhary, Narayan and Girish Nath Jha. (2011). *Creating multilingual parallel corpora in Indian Languages*. In: Proceedings of the 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC '11), Poznan, Poland, pp. 85-89.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li and Wim Peters. (2011). *Text Processing with GATE (Version 6)*. http://tinyurl.com/gatebook.

Jha, Girish Nath.(2009). *Indian Language Corpora Initiative (ILCI)*. Invited talk, 4th International Language and Technology Conference (4th LTC), Poznan, Poland, (Nov 6, 2009).

Jha, Girish Nath. (2010) . *The TDIL program and the Indian Language Corpora Initiative (ILCI)*. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), pp. 982-985.

Kaplan, D., Iida, R. and Tokunaga, T. (2010). *Annotation Process Management Revisited*. In: Proceedings of the Seventh conference on International Language

Resources and Evaluation (LREC'10), pp. 3654-3661.

Russell, B. C., A. Torralba, K. P. Murphy, and W. T. Freeman. (2005). *Labelme: A database and web-based tool for image annotation*. Technical report, Tech. Rep. MIT-CSAIL-TR-2005-056, Massachusetts Institute of Technology.

Singh,D..P. (2010). *Comparative Study of Hindi Part of Speech Tagsets*. Unpublished M.Phil dissertation. Jawaharlal Nehru University, New Delhi, India.

Stuhrenberg, Maik, Daniela Goecke, Nils Diewald, Alexander Mehler, and Irene Cramer. (2007). *Web-based annotation of anaphoric relations and lexical chains.* In Proc. of the Linguistic Annotation Workshop, pp. 140–147, Prague, Czech Republic, June. Association for Computational Linguistics.

Dependency Bank

Hans Martin Lehmann, Gerold Schneider
English Department
University of Zurich
E-mail: gschneid@es.uzh.ch, hmlehman@es.uzh.ch

Abstract

In this paper we present a dependency bank framework that scales from small sets like the ICE corpora to data sets of more than 1000 million words. The dependency bank encodes information at the levels of word-class, chunking and dependency syntax. We discuss the structure of the database, the annotation chain and present a web-based interface. We then discuss potential applications as well as limitations of our fully automatic annotation strategy.

Keywords: dependency bank, automatic syntactic annotation, corpus query tool

1 Introduction

In recent years, parsing technology has made considerable advances, opening new perspectives for descriptive linguistics. Van Noord and Bouma (2009, 37) state that "[k]nowledge-based parsers are now accurate, fast and robust enough to be used to obtain syntactic annotations for very large corpora fully automatically." We apply parsed corpora as a new resource for linguists.
In this paper we present a dependency bank framework that scales from small sets like the ICE corpora to data sets of more than 1000 million words. The dependency bank encodes information at the levels of word-class, chunking and dependency syntax. We discuss the structure of the database, the annotation chain and present a web-based interface. We then critically discuss applications as well as the limitations of our fully automatic annotation strategy.

2 Annotation

We have developed an annotation chain using robust state-of-the-art tagging, lemmatizing, chunking and parsing tools, which feeds the annotation into a series of SQL databases. Up to and including the input for the parser, the annotated data of our currently most used chain is in XML. This chain uses the C&C tagger, the *morpha* lemmatizer and the LT-TTT2 chunker (Grover 2008). We also have an annotation chain which uses Treetagger (Schmid 2008) for tagging and lemmatizing, and Carafe[1] for chunking. The output of LT-TTT2 is given in figure 1.
For the syntactic annotation, we use Pro3Gres (Schneider 2008), a dependency parser. Dependency Grammar goes back to Tesnière (1959) and is used by many parsers (e.g.

```
<?xml version="1.0"?>
<text>
  |  <p>
  |   |  <s id="s1">
  |   |   |  <ng>
  |   |   |   |  <w pws="yes" id="w1" p="DT">A</w>
  |   |   |   |  <w pws="yes" id="w3" p="JJ">long-term</w>
  |   |   |   |  <w l="borrower" pws="yes" id="w13" p="NN" headn="yes">borrower</w>
  |   |   |  </ng>
  |   |   |  <vg tense="pres" voice="act" asp="simple" modal="no">
  |   |   |   |  <w l="have" pws="yes" id="w22" p="VBZ" headv="yes">has</w>
  |   |   |  </vg>
  |   |   |  <w pws="no" id="w25" p=",">,</w>
  |   |   |  <rg>
  |   |   |   |  <w pws="yes" id="w27" p="RB">therefore</w>
  |   |   |  </rg>
  |   |   |  <w pws="no" id="w36" p=",">,</w>
  |   |   |  <vg tense="inf" voice="act" asp="simple" modal="no">
  |   |   |   |  <w pws="yes" id="w38" p="TO">to</w>
  |   |   |   |  <w l="pay" pws="yes" id="w41" p="VB" headv="yes">pay</w>
  |   |   |  </vg>
  |   |   |  <ng>
  |   |   |   |  <w l="lender" pws="yes" id="w45" p="NNS" headn="yes">lenders</w>
  |   |   |  </ng>
  |   |   |  <ng>
  |   |   |   |  <w pws="yes" id="w53" p="DT">a</w>
  |   |   |   |  <w l="premium" pws="yes" id="w55" p="NN" headn="yes">premium</w>
  |   |   |  </ng>
  |   |   |  <w pws="no" id="w62" p=",">,</w>
  |   |   |  <pg>
  |   |   |   |  <w pws="yes" id="w64" p="IN">for</w>
  |   |   |  </pg>
  |   |   |  <ng>
  |   |   |   |  <w vstem="lose" l="loss" pws="yes" id="w68" p="NN" headn="yes">loss</w>
  |   |   |  </ng>
  |   |   |  <pg>
  |   |   |   |  <w pws="yes" id="w73" p="IN">of</w>
  |   |   |  </pg>
  |   |   |  <ng>
  |   |   |   |  <w l="equity" pws="yes" id="w76" p="NN" headn="yes">equity</w>
  |   |   |  </ng>
  |   |   |  <w sb="true" pws="no" id="w82" p=".">.</w>
  |   |  </s>
  |  </p>
</text>
```

Figure 1. LT-TTT2 output for BNC:K92:1437.

Tapanainen and Järvinen 1997, Nivre 2006). Pro3Gres is implemented in Prolog. It uses a hand-written grammar, which models linguistic competence, and statistical disambiguation, which models performance. The parser learns the performance statistics from the Penn Treebank. The performance model measures attachment probabilities for a dependency relation, given the lexical heads of the governor and the dependent. Figure 1 shows the sample output from LT-TTT2 for a sentence from the BNC. The parser takes the chunked sentence as input, numbers the chunks and annotates them with dependency relations as illustrated in figure 2.

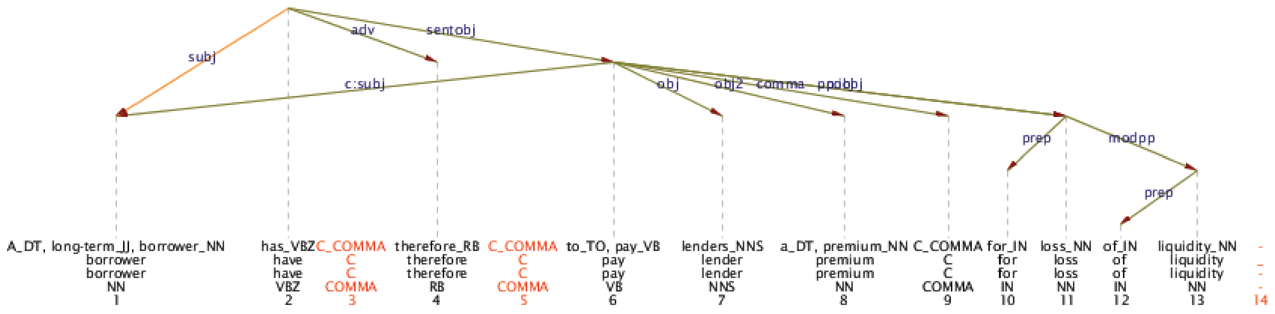

subj
adv
sentobj
c:subj
obj
obj2
comma
ppobj
prep
modpp
prep
A_DT, long-term_JJ, borrower_NN borrower borrower NN 1
has_VBZ have have VBZ 2
C_COMMA C C COMMA 3
therefore_RB therefore therefore RB 4
C_COMMA C C COMMA 5
to_TO, pay_VB pay pay VB 6
lenders_NNS lender lender NNS 7
a_DT, premium_NN premium premium NN 8
C_COMMA C C COMMA 9
for_IN for for IN 10
loss_NN loss loss NN 11
of_IN of of IN 12
liquidity_NN liquidity liquidity NN 13
- - - - 14

Figure 2. Dependency graph of a BNC sentence (BNC:K92:1437) by the Pro3GRes parser

[1] http://sourceforge.net/projects/carafe/

The parser is fast and robust and has state-of-the-art performance, as we discuss in the following. The dependency format of the parser is similar to GREVAL, a parser-internal conversion to the Stanford Scheme (de Marneffe 2006) is included (Haverinen et al. 2008), parser-external conversions to the CoNLL format (Nivre 2007) have been made (Schneider et al. 2007).

2.1 Performance of the Parser

We have evaluated Pro3Gres on various genres. We have used the GREVAL corpus (Carroll 2004), biomedical texts, the BNC, texts from the International Corpus of English (ICE) and the historical Archer Corpus.

Table 1. Evaluations of Pro3Gres parser.

GREval	Subj.	Obj.	N-PP	V-PP	sub.clause
Precision	92%	89%	74%	72%	74%
Recall	81%	84%	65%	85%	62%
GENIA	Subj.	Obj.	N-PP	V-PP	sub.clause
Precicion	90%	93%	85%	82%	
Recall	87%	91%	82%	84%	
BNC W	Subj.	Obj.	N-PP	V-PP	sub.clause
Precision	86%	87%		89%	
Recall	83%	88%		70%	
BNC X	Subj.	Obj.	N-PP	V-PP	sub.clause
Precision	89%	75%	75%	83%	73%
Recall	86%	83%	77%	69%	63%

Table 1 shows evaluations of recall and precision for several sets of data and different versions of the annotation chain. GREval (Carroll et al. 2003) is a manually annotated evaluation corpus of 500 sentences from the Susanne corpus. GENIA contains 100 manually annotated random sentences from the biomedical domain (Kim et al. 2003). We have manually annotated 100 sentences from two different pipelines of our system. BNC X(ML) contains 100 random sentences from the pipeline used in this paper. BNC World contains 100 different random sentences from an earlier version of the BNC corpus and a pipeline that is based on a different tagger and a different chunker. The evaluation differences between the two different BNC pipelines are partly random fluctuations as we have used a different 100 random sentence test set. Error sources include attachment errors but also tagging, chunking and lemmatization errors.

2.2 Strategies for scaling to a 1 billion word corpus

On a typical multi-core server, the BNC (Aston & Burnard 1998) now parses in under 24 hours. Depending on their availability and workload, we parse our 1000 million word set with 106 threads in about 36 hours in a parallel architecture. We have met a number of challenges until we managed to scale up to such large amounts of texts. In the following, we describe the strategies that we have employed during parser development. Except for the last two, these solutions are described in Schneider (2008). Pro3Gres has been specifically developed to be a robust parser.

Part-Of-Speech Tagging: Like many parsers, we use tagging as a finite-state technology pre-processing step. Prins (2005, 72-74) reports that tagging preprocessing parsers are up to 10 times faster, and that the accuracy increases slightly if reasonable filtering parameters are used. Kaplan et al. (2004) describe the integration of finite-state morphology and part-of-speech tagging as an essential step for the development of truly broad-coverage grammar and robust parsers.

Base Phrase Chunking: Using finite-state approaches instead of expensive full parsing reduces processing time. Prins (2005) shows that chunking preprocessing in parsing generally leads to a moderate increase in parsing speed. The integration of chunking allows one to parse only between the heads of chunks (Abney, 1996).

Dependency Grammar: The integration of chunking and parsing fits Dependency systems particularly well, as Abney (1996) point outs. Tesnière's concept of nucleus largely corresponds to chunks. Dependency Grammar has the practical advantages that trees are less neseted (fewer reduction steps), and that there are no empty nodes. The latter allows one to use efficient parsing alogrithms like CYK.

The CYK Parsing Algorithm: The CYK algorithm (Younger 1967) has complexity n^3, which is relatively low.

Beam Search and Pruning: In long sentences, the search spaces still get out of hand. We use a beam search which typically keeps a maximum of 5-10 variants per span. Very unlikely local analyses regularly get pruned. Pruning is partly complexity-dependent: thresholds increase as a function of the number of entries in the parsing chart from a certain threshold on. In practice, charts with more than a few thousand entries get rare due to this approach.

Last Resort Time-Outs: There are few real world sentences which are several hundred words long. They usually contain large lists in sentence form. The longest sentence in the BNC is over 90 chunks long. While typical sentences take only a fraction of a second to parse, such sentences can take several hours to parse. For practical reasons, we maximally allow 5 minutes per sentence. 55 sentences in the BNC are affected by this time-out.

Parallelisation with X-grid: We use an Xgrid parallel architecture[2] in our annotation chain, which distributes the corpora to several servers. Depending on their availability and workload, we apply the full annotation chain with tagging, lemmatization, chunking and parsing to our 1000 million word set with 102 threads in about 42 hours.

2.3 Text Selection and Collection

Our collection of texts amounts to 1,063,921,539 running words, punctuation excluded. Collecting more than a billion running words requires electronic sources. Using the web as a corpus is an obvious strategy. Keller and Lapata (2003), Lapata and Keller (2005) and Evert (2010) show that, for many tasks, larger less carefully sampled web corpora are inferior to slightly smaller carefully sampled corpora. We have decided to collect specific sources of data from the web. The vast majority

2

https://developer.apple.com/hardwaredrivers/hpc/xgrid_intro.html of our data is news related. The CNN Transcripts make up about half of our data. The CNN data ranges from written to be spoken to fairly spontaneous conversation produced for a TV audience. The written news texts are derived from newspapers. The New York Times data is collected from the web, whereas the other newspaper data are derived from the editorial content for the year 1999 acquired on CD Rom. The exact word counts are given in table 2.

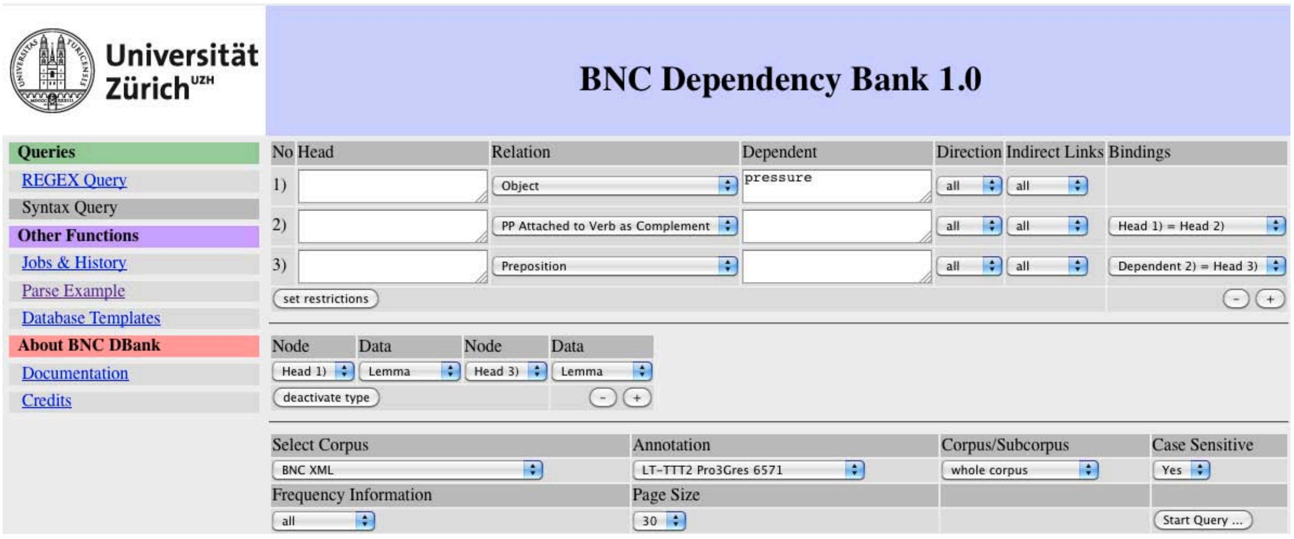
Table 2. Word count per source.

Source	Short ref	Words
CNN Transcripts 03-11	cnn0311	528179519
New York Times 06-11	nyt0611	254063861
BNC written	bncxwri	92519252
Los Angeles Times	latm	44312188
Boston Globe	bogl	39378716
The [London] Times	tlnd	38264171
The Daily Mail	tdma	23120176
USA Today	usat	19032748
Times Sunday	tlns	18457212
The Mail on Sunday	tmos	6593696

We have decided to add the written component of the British National corpus, to lessen the bias toward US data in the written material, and because it is particularly carefully sampled (Evert, 2010). This collection of data limits the type of research questions that can sensibly be investigated. Many research questions will be answerable with the help of smaller, more carefully sampled data like the ICE corpora or the BNC corpus. For an application of the current framework to the BNC see Lehmann and Schneider (forthcoming) and for the ICE corpora Lehmann and Schneider (in preparation) as well as the Dependency Bank website (http:/www.es.uzh.ch/dbank). The investigation of lexis-syntax interactions is an area where the large amount of data makes a difference that outweighs the advantages of smaller more carefully sampled corpora.

3 Coping with the annotated data

The output of the dependency parser Pro3Gres is originally in a Prolog predicate-argument format. We have conducted initial experiments with the Prolog output in a Prolog database. This allows for a flexible, powerful and intuitive query language, and the approach is reasonably fast for corpora of one to ten million words. But it failed to scale to 100 million word corpora like the BNC. The standard Prolog settings to keep all data in memory, as well as standard Prolog indexing reach their limits. Our set of more than one billion running words is at least two orders of magnitude outside the range of an approach based on prolog.
Processing the one billion words with the latest version of our annotation pipeline produces a flood of 1.2 TB of data, containing 63,299,067 s-units, 824,660,963 chunks and 966,854,086 dependency relations, of which 676,948,105 are chunk external and 289,905,981 are chunk internal.
We use MySQL, a relational SQL database system for structured storage and analysis of the annotated data. From the original prolog format, the material is translated into a set of four tables.
A results table with one record per s-unit, where we store the different levels of annotation, the raw parser output, pointers to the source of the s-unit as well as keys for the association of meta-data.
A syntax table containing one record per dependency relation indicating the label, direction, direct or indirect attachment as well as an identifier and the lemma for the head and the dependent of every dependency relation.
Via the identifier, additional information for each chunk is available in a chunk table. In addition to the lemma, the chunk table provides word-form, word-class and additional features like tense, voice, aspect, number and the presence of modals or negation produced by LT-TTT2. We also store information about the number and type of dependency relations that lead to or originate in the node.


Universität Zürich UZH

BNC Dependency Bank 1.0

Queries
REGEX Query
Syntax Query
Other Functions
Jobs & History
Parse Example
Database Templates
About BNC DBank
Documentation
Credits

No	Head	Relation	Dependent	Direction	Indirect Links	Bindings
1)		Object	pressure	all	all	
2)		PP Attached to Verb as Complement		all	all	Head 1) = Head 2)
3)		Preposition		all	all	Dependent 2) = Head 3)

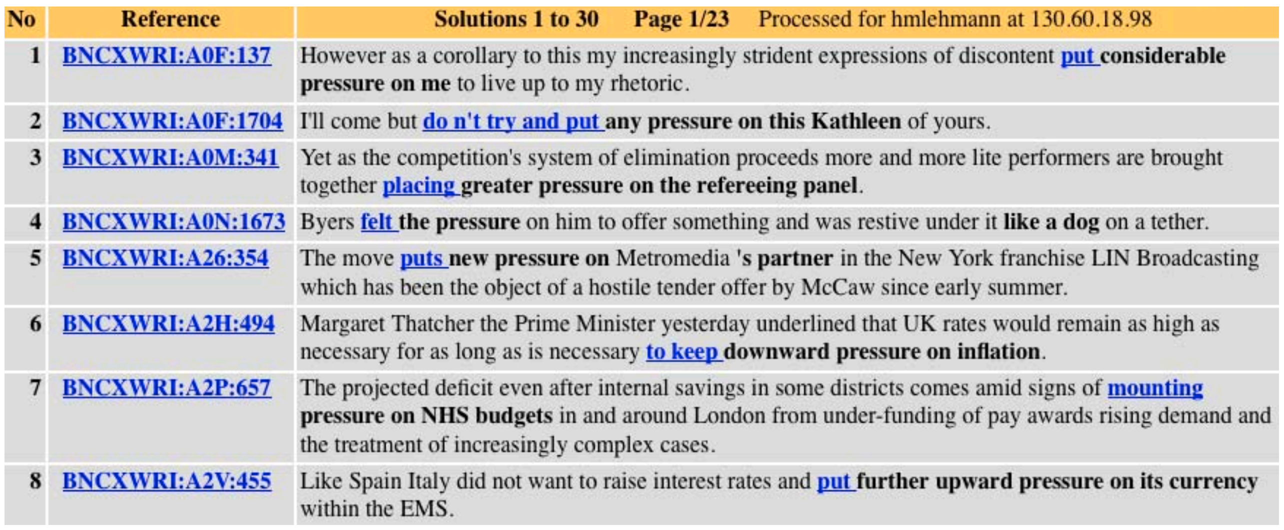
set restrictions − +

Node	Data	Node	Data
Head 1)	Lemma	Head 3)	Lemma

deactivate type − +

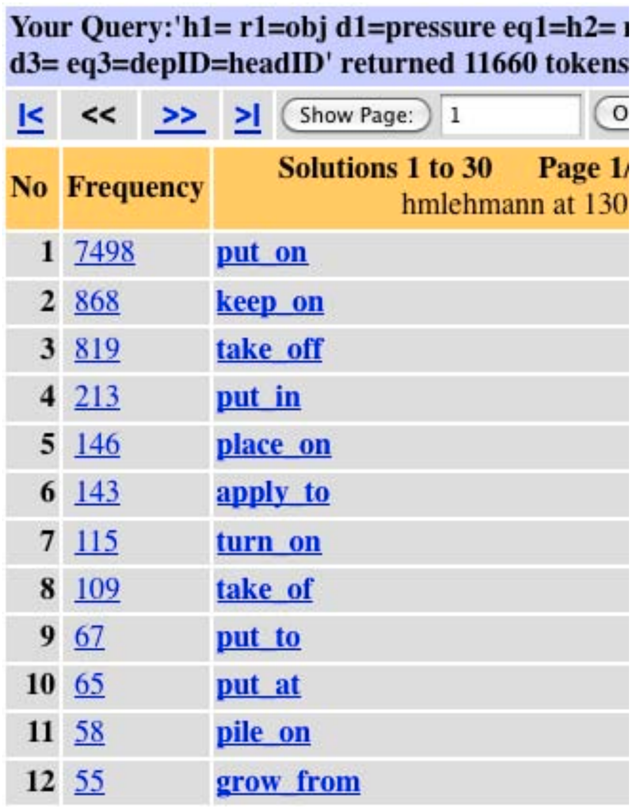
Select Corpus	Annotation	Corpus/Subcorpus	Case Sensitive
BNC XML	LT-TTT2 Pro3Gres 6571	whole corpus	Yes
Frequency Information	**Page Size**		
all	30		Start Query ...

Figure 3. Sample query on the BNC part of the Dependency Bank



No	Reference	Solutions 1 to 30 Page 1/23 Processed for hmlehmann at 130.60.18.98
1	BNCXWRI:A0F:137	However as a corollary to this my increasingly strident expressions of discontent **put considerable pressure on me** to live up to my rhetoric.
2	BNCXWRI:A0F:1704	I'll come but **do n't try and put any pressure on this Kathleen** of yours.
3	BNCXWRI:A0M:341	Yet as the competition's system of elimination proceeds more and more lite performers are brought together **placing greater pressure on the refereeing panel**.
4	BNCXWRI:A0N:1673	Byers **felt the pressure** on him to offer something and was restive under it **like a dog** on a tether.
5	BNCXWRI:A26:354	The move **puts new pressure on** Metromedia **'s partner** in the New York franchise LIN Broadcasting which has been the object of a hostile tender offer by McCaw since early summer.
6	BNCXWRI:A2H:494	Margaret Thatcher the Prime Minister yesterday underlined that UK rates would remain as high as necessary for as long as is necessary **to keep downward pressure on inflation**.
7	BNCXWRI:A2P:657	The projected deficit even after internal savings in some districts comes amid signs of **mounting pressure on NHS budgets** in and around London from under-funding of pay awards rising demand and the treatment of increasingly complex cases.
8	BNCXWRI:A2V:455	Like Spain Italy did not want to raise interest rates and **put further upward pressure on its currency** within the EMS.

Figure 4. The first hits for the query given in figure 3.

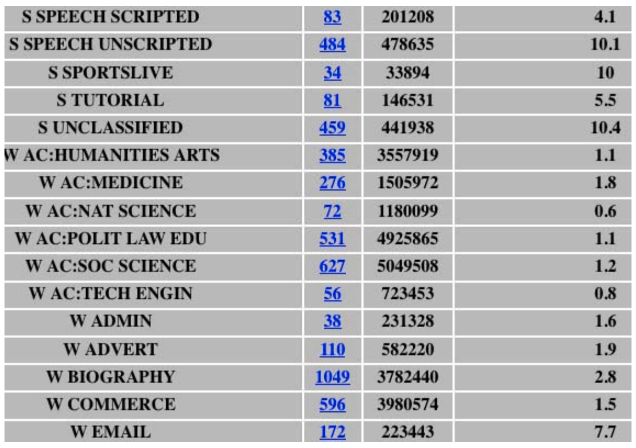


Your Query:'h1= r1=obj d1=pressure eq1=h2=
d3= eq3=depID=headID' returned 11660 tokens

No	Frequency	Solutions 1 to 30 Page 1/ hmlehmann at 130
1	7498	put_on
2	868	keep_on
3	819	take_off
4	213	put_in
5	146	place_on
6	143	apply_to
7	115	turn_on
8	109	take_of
9	67	put_to
10	65	put_at
11	58	pile_on
12	55	grow_from

Figure 5. List of verb-preposition types ordered frequency.

Where available, a fourth database contains meta-data, like spoken or written, text-type, region, speaker age, word-frequencies etc. The present setup permits direct querying of the annotated data via SQL queries. The central starting point for querying the database is the syntax table. For complex queries involving several dependencies we join the syntax table with itself. Not surprisingly, disk access turned out to be the limiting factor for such queries. We have optimized access with several indexes for the syntax table. In addition we make use of SSDs configured in RAID 0 to improve disk in-out. A fairly large subset of queries performs fast enough



S SPEECH SCRIPTED	83	201208	4.1
S SPEECH UNSCRIPTED	484	478635	10.1
S SPORTSLIVE	34	33894	10
S TUTORIAL	81	146531	5.5
S UNCLASSIFIED	459	441938	10.4
W AC:HUMANITIES ARTS	385	3557919	1.1
W AC:MEDICINE	276	1505972	1.8
W AC:NAT SCIENCE	72	1180099	0.6
W AC:POLIT LAW EDU	531	4925865	1.1
W AC:SOC SCIENCE	627	5049508	1.2
W AC:TECH ENGIN	56	723453	0.8
W ADMIN	38	231328	1.6
W ADVERT	110	582220	1.9
W BIOGRAPHY	1049	3782440	2.8
W COMMERCE	596	3980574	1.5
W EMAIL	172	223443	7.7

Figure 6. Distribution according to genre labels.

for an interactive system.

4 Querying: the Interface

We have developed a web-based interface to the dependency-parsed database. The interface offers syntactic queries with and without lexical constraints. It also provides a tool for the analysis of lexical and grammatical types defined by underspecified queries. Results sets can be distributed over the categories defined by the meta-data, resulting in tabulations and cross-tabulations of raw and relative frequencies. The query window has four parts: the **subgraph query** at the top, where arbitrarily complex subtree queries can be composed; the **morphosyntactic restrictions** in the middle, where lemma, tag, voice, aspect etc. can be restricted or collected by type (the **type specifications**); and the **corpus selection** at the bottom.

In the following, we give a brief example of a simple query. The query given in figure 3 reports verb constructions with *pressure* as object that also attach a prepositional phrase. Each row in the query window represents a dependency relation. In **subgraph query** row 1) we specify that *pressure* is object of any verb

(Head 1), in row 2) we demand that the same verb (Head1=Head2) attaches an oblique NP. In row 3) we specify the preposition depending on the oblique NP.
Below the dependency specification, we set a **type specification**. We instruct the system to collect the lemmas of the verb (Head 1) and the preposition (Dependent 3). In the **corpus selection** part, we can select the corpus to be searched. With cold caches, this query executes in about 4 seconds on the BNC, and less than a minute on the billion word data set. Response times depend on the number of specified dependency rows as well as on the dependency with the lowest frequency. Specifying rare lemmas typically reduces processing time massively.
The first hits reported are given in figure 4. Figure 5 shows a frequency list of the specified types. The types are linked to the source sentences, which can be inspected by clicking on them.

Figure 6 illustrates the possibility of distributing result sets over the meta-information. It is based on a query for all verbal particles in the written BNC, for which rich contextual information is available. Figure 6 shows a tabulation according to genres. The genre is in the first column, absolute counts in column 2, and frequency per 10.000 words in the fourth column. Relative frequencies for arbitrary classifications are calculated on the fly. We can observe, that Spoken (S) has considerably higher counts than written (W). In scientific texts, verbal particles are virtually absent, only in written emails they reach levels that are comparable to spoken language.

5 Linguistic Applications

For the automatically annotated dependency bank we see two main areas of application: The investigation of the lexis-syntax interface and the explorative analysis of varieties.
Only few currently available English corpora are manually analyzed for syntactic structure, for example, ICE-GB and the Penn Treebank. However, they are too small for the study of typical lexis-syntax interactions, where we cross-tabulate syntactic phenomena over the large number of lexical items.
With the help of the dependency bank, we have investigated the lexis-syntax interface in the active-passive alternation (Lehmann and Schneider 2010), in verb attached PP structures (Lehmann and Schneider 2011) and the dative shift alternation (Lehmann and Schneider, in press).
Currently, there are no treebanks covering English regional varieties. Here, automatically parsed corpora can be used as a stopgap to manually annotated Treebanks. We have annotated the regional set of ICE corpora as well as the diachronic Archer and ZEN corpora. Based on these data sets we have described language variation, for example the diachronic development of relative clauses (Hundt et al. 2011) or verb-preposition structures in world Englishes (Schneider and Zipp, accepted for publication).

As a matter of course, automatic parsing is not error-free. However, the study of lexis-syntax interface would not be possible without the large amount of data available via automatic annotation. In addition, the error rate is sufficiently low for the explorative analysis of varieties of English.

6 Related Approaches

The last decades have seen the emergence of many fast and robust parsers. The easily accessible head-head relationships make dependency parsers particularly suitable for IR applications and lexis-grammar research. Well-known dependency parsers include Nivre (2006), the systems that have participated in the CoNLL shared task (Nivre et al. 2007) as well as the Stanford parser (de Marneffe 2006). We find that the Stanford parser uses a linguistically particularly convincing scheme that is relatively close to ours and can be mapped at runtime (Haverinen et al 2008). In comparison to Pro3Gres, the Stanford parser is relatively slow for parsing large corpora, however.
Unlike these robust statistical parsers, Pro3Gres uses a hand-written grammar that can be modified for linguistic experiments, and inspected by the interested linguist. Together with the performance, this has made it possible to produce parses with a grammar modified for specific research questions (Lehmann and Schneider in press).
Several large-scale parsing projects have been presented, for example Andserson et al. (2008) who have parsed the BNC with the RASP parser (Briscoe et al. 2006).
Interfaces for querying large corpora have recently become available. Ghodke and Bird (2010) present an interface for querying datasets of up to 26 million sentences, parsed in Penn Treebank format. This corresponds to about half a billion words. They use Tgrep2 as a query language, whereas we provide a graphical query builder which non-computational linguists may find easier to use.

7 Conclusion

In this paper we have given an overview of the process of annotating a one billion word set of data with the help of Pro3Gres, a dependency parser. We have presented the a dependency bank framework as a new resource. We have outlined the functionality provided by our web-based interface to the Dependency Bank. We have found that our approach scales to corpora of more than one billion words.
There are many different directions in which we plan to further develop the dependency bank framework. An interesting option to explore is the improvement of the parser with the help of its own output. The very accessibility of the data may be used to revise, adapt and improve the parser. There are many possible additions to the functionality of the web-based interface. Currently, we offer only frequency-based analyses of specified types. We envisage a system that permits a more sophisticated approach based on various measures of surprise.

8 References

Andersen, Ø. E.; Nioche, J.; Briscoe, T.; Carroll, J. (2008). The BNC Parsed with RASP4UIMA. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco.

Aston, G.; Burnard, L (1998). *The BNC Handbook. Exploring the British National Corpus with SARA*. Edinburgh: Edinburgh University Press.

Briscoe, E.; Carroll, J.; Watson, R. (2006) The Second Release of the RASP System. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia.

Carroll, J.; Minnen, G.; and Briscoe, E. (2003). Parser evaluation: using a grammatical relation annotation scheme. In Anne Abeillé, editor, Treebanks: *Building and Using Parsed Corpora.* Dordrecht: Kluwer, 299–316.

de Marneffe, M.-C.; MacCartney, B.; Manning, C.D. (2006). Generating typed dependency parses from phrase structure parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Evert, S. (2010). Google Web 1T5 n-grams made easy (but not for the computer). In *Proceedings of the 6th Web as Corpus Workshop (WAC-6)*, Los Angeles, CA.

Ghodke, S.; Bird, S. (2010). Fast query for large treebanks. In *Human Language Technologies: Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, USA.

Grover, C. (2008). *LT-TTT2 Example Pipelines Documentation*. Edinburgh Language Technology Group, July 24.

Haverinen, K.; Ginter, F.; Pyysalo, S.; Salakoski,T. (2008). Accurate conversion of dependency parses: targeting the Stanford scheme. In Proceedings of Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland.

Hundt, M.; Dension, D.; Schneider, G. (2011). Retrieving relatives from historical data. *Literary and Linguistic Computing*.

Kaplan, R. M.; Maxwell III, J.T.; King, T.; Crouch, R. (2004). Integrating finite-state technology with deep LFG grammars. In ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP (ComShaDeP 2004), Nancy, France.

Keller, F.; Lapata, M. (2003). Using the web to obtain frequencies for unseen bigrams. Computational Linguistics, 29:3:459–484.

Kim, J. D.; T. Ohta, Tateisi, Y.; Tsujii,J. (2003). GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(1):180–182.

Lapata, M.; Keller, F. (2005). Web-based models for natural language processing. ACM Transactions on Speech and Language Processing, 2:1:1–31.

Lehmann, H.-M.; Schneider,G. (2010). Parser-Based Analysis of Syntax-Lexis Interaction. In A. H. Jucker, D. Schreier & M. Hundt, (Eds,), *Corpora : Pragmatics and discourse : papers from the 29th International conference on English language research on computerized corpora (ICAME 29), Ascona, Switzerland, 14-18 May 2008* (Language and computers 68). Amsterdam : Rodopi. 477-502.

Lehmann, H.-M.; Schneider, G. (2011). A large-scale investigation of verb-attached prepositional phrases. In S. Hoffmann, P. Rayson, & G. Leech, (Eds.), *Studies in Variation, Contacts and Change in English, Volume 6: Methodological and Historical Dimensions of Corpus Linguistics*. Helsinki: Varieng.

Lehmann, H.-M.; Schneider, G. (in press). Syntactic Variation and Lexical Preference in the Dative-shift Alternation. In J. Mukherjee & M. Huber, (Eds.), *Corpus Linguistics and Variation in English: Theory and Description*. Amsterdam: Rodopi.

Nivre, J. (2006). Inductive Dependency Parsing. Text, Speech and Language Technology 34. Springer, Dordrecht, The Netherlands.

Nivre, J.; Hall, J.; Kübler, S.; McDonald, R.; Nilsson, J.; Riedel, S.; Yuret,D. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Prins, R. (2005). Finite-State Pre-Processing for Natural Language Analysis. Ph.D. thesis, Behavioral and Cognitive Neurosciences (BCN) research school, University of Groningen.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*.

Schneider, G.; Kaljurand, K; Rinaldi, F.; Kuhn,T. (2007). Pro3Gres parser in the CoNLL domain adaptation shared task. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1161–1165, Prague.

Schneider, G. (2008). *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis. Institute of Computational Linguistics, University of Zurich.

Schneider, G.; Zipp,L. (accepted for publication). Discovering new verb-preposition combinations in New Englishes. *Corpus Linguistics and variation in English: Focus on Non-native Englishes*. Helsinki: Varieng.

van Noord, G.; Bouma, G. (2009). Parsed corpora for linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, p. 33–39, Athens, Greece. Association for Computational Linguistics.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. Information and Control, 10:189-208.

Introducing the CLARIN-NL Data Curation Service

Nelleke Oostdijk and Henk van den Heuvel

CLS/CLST (Centre for Language and Speech Technology)
Radboud University Nijmegen
P.O. Box 6500 HD Nijmegen, The Netherlands
E-mail: {n.oostdijk|h.vandenheuvel}@let.ru.nl

Abstract

In this paper we introduce the CLARIN-NL Data Curation Service. We highlight its tasks and its mediating position between researchers and the CLARIN Data Centres. We outline a scenario for successful data curation and stress the need to take notice of the factors that determine the desirability and feasibility of data curation. Finally, we present and discuss an exemplary case that illustrates the relevant issues involved in setting up a data curation plan.

Keywords: data curation, corpus management, sustainable infrastructure.

1. Introduction

Following decades in which a great deal of effort was spent on the creation of resources, currently there are several initiatives worldwide that aim to create an interoperable, sustainable research infrastructure. Examples, more specifically for the arts and humanities, are the US project Bamboo[1] and the European CLARIN initiative.[2] An integral part of such an infrastructure constitute the resources (data and tools) which researchers in the various disciplines employ. Whether the infrastructure will be successful in supporting the needs of the research communities it intends to cater for, depends on a number of factors. One factor is that resources that are or could be relevant to the wider research community are made visible through this infrastructure and, to the extent possible, accessible and usable.

Over the past decades numerous datasets have been collected and annotated by researchers for use in their own research. Often such data sets sank into oblivion once the research results had been published, while occasionally data were actually lost. With the years it has become apparent that unless appropriate action is undertaken to actively *curate* existing resources, many are at the risk of being lost as individual researchers or research groups often lack the expertise and the means to take the necessary measures to ensure their future availability.

By resource curation we mean the planning, allocation of financial and other means, and application of preservation methods and technologies to ensure that digital information of enduring value remains accessible and usable. It encompasses material that begins its life in digital form as well as material that is converted from traditional analog to digital formats. Digital information must be stored long-term and error-free, with means for retrieval and interpretation, for the entire time span the information is required for; in other words, it must be possible to decode and transform the retrieved files – of texts, charts, images or sound - into usable representations (cf. Hedstrom 1997).

Resource curation is important

- from an economic point of view;
  Curation is needed to prevent loss of resources that were created at substantial efforts and expenses. Loss may occur as a result of media deterioration or digital obsolescence. Costs may incur when resources are lost and resources must be rebuilt. In some cases, resources are unique and cannot be replaced if destroyed or lost.
- in terms of scientific interest;
  Curation grants access to the resources to a wider user community, allowing researchers to share access to data sets and permit replicability in research.
- for reasons of cultural heritage.

The structure of the present paper is as follows. In Section 2 the objectives and the background for setting up the CLARIN-NL data curation service (DCS) are described. Section 3 focuses on the positioning of the DCS in the language resources infrastructure context in the Netherlands and the tasks with which it has been charged. In Section 4 we report on experiences with the curation by the DCS of three data collections, viz. the Dutch Bilingual Database, Roots of Ethnolects and TCULT. Section 5 concludes the paper.

2. Background

In The Netherlands CLARIN-NL[3] (Odijk 2010) received funding from the Dutch government for implementing a programme that would contribute to the development of a sustainable research infrastructure for the humanities and linguistics in particular. CLARIN-NL is carried out jointly by technical specialists, technology providers and researchers. An effort is made to involve the intended

[1] www.projectbamboo.org
[2] www.clarin.eu
[3] www.clarin.nl

users through various application projects in which local repositories are integrated and local services set up for prototypical test installations as initial demonstrators, enabling evidence-based contributions to the discussion on standards and best practices for inter-operability, and to contribute to the survey of requirements for the infrastructure technology.[4]

From the start of the programme (2009), in CLARIN-NL funding has been available for projects directed at resource curation. Although a number of curation projects were undertaken, the calls for proposals have been less successful in reaching resource producers and owners who were not already aware of and/or participating in CLARIN-NL. In October 2010 the CLARIN-NL Executive board Board therefore initiated a pilot project that should investigate the need and possibility for establishing a Data Curation Service (DCS) task force that would salvage valuable corpora and data sets that are at the risk of being lost. The idea was that a dedicated team of specialists should be made responsible for curating data residing with humanities researchers, especially those who are reluctant or incapable of undertaking the curation themselves. In such a scenario curation is carried out with minimal support from the original researcher who created, owns and/or manages the data. The data would subsequently be made available to the CLARIN community through one of the CLARIN-NL Data Centres (Odijk 2010).

The pilot project was carried out between 1 November 2010 and 1 February 2011. The aim was to establish whether there was a sufficient basis to assume that such a service would meet with a demand in the field and to develop ideas about the form such a service should be take, and also the effort and expertise required.

In the pilot project various data curation models and frameworks (such as the DCC Curation Lifecycle Model[5]) were investigated and the data curation policies adopted by other parties (such as libraries and archives, including for example the National Library of the Netherlands and Data Archiving and Networked Services (DANS) were looked into. Moreover, the project charted the role of various stakeholders (e.g. researchers, research institutes but also funding agencies like NWO) and organizations such as SURF and the Dutch Language Union.

As regards the needs and the priorities in curating resources, the roadmaps and surveys compiled by ELSNET and the Dutch Language Union provided pertinent information. Consultation of the national research database maintained by the Royal Netherlands Academy of Sciences (KNAW) served to identify which resources feature(d) in current or recent humanities research. Criteria were formulated for prioritizing resources for curation.

The main findings obtained in the pilot project were summarized in a report (Oostdijk 2011). As it was found that there was indeed a need and a basis for a DCS task force, CLARIN-NL in September 2011 decided to establish the DCS at CLST in Nijmegen.

3. The CLARIN-NL DCS

3.1 Position and tasks

The DCS has been operational since January 2012. It aims to contribute the research infrastructure that CLARIN is implementing by salvaging resources and advising on best practices and the use of standards. Set up as a service, the DCS maintains close contact with the research communities as a mediator between these and the CLARIN Data Centres. The DCS prepares resources for archiving at the Data centres, but does not archive any resources itself.

Accordingly, the tasks of the DCS are defined as follows:

1. Curation of resources, especially those presently held by individual researchers or research groups
2. Assisting in the curation efforts of CLARIN Data Centres (if and when such is desired)
3. Advising researchers who wish to undertake the curation of their resources themselves

The curation of resources held by individual researchers or research groups form the core of the work undertaken by the DCS. As the DCS receives funding from CLARIN-NL, efforts are directed primarily at language resources stored and used in The Netherlands. The final decision to curate a resource is made by CLARIN-NL's Executive Board, based on a proposal by the DCS.

3.2 Data curation

The tasks and actions involved in the curation of resources are summarized in Figure 1.

A first step towards curation is the identification and assessment of candidate resources. This may require a great deal of effort, both in terms of the time and the persistence needed for tracking down the resource and whatever relevant information there is. It is a critical step in the curation process as it should result in a go or no-go for moving ahead with the drawing up of a plan for actually curating the resource. The work undertaken as part of Task A should prevent money and effort going to waste in failing curation efforts. Task A is ideally carried out in close collaboration with the resource owner/producer.

The assessment of a candidate resource considered for curation concerns two aspects: it should be established whether it is desirable to have the resource curated and whether indeed successful curation is feasible.

Whether it is desirable to curate a resource (Action A2) is not a question that can be answered straightforwardly, as various factors need to be considered:

[4] Cf. CLARIN-NL Long Term Programme 2009-2014. Retrievable from http://www.clarin.nl/system/files/CLARIN-NL%20Multiyear%20Programme%20090409-2.pdf

[5] http://www.dcc.ac.uk/resources/curation-lifecycle-model.

Task A. Identification and assessment
Actions 1. Identify candidate resources; collect info as to a. the owner/producer b the type of resource c. the licensing restrictions/conditions d. the size e. the format(s) f. the metadata available g. the nature of enrichment/annotations etc. 2. Assess the desirability of curation 3. Assess the feasibility of successful curation
Task B. Development of a curation plan
Actions 4. Evaluate the content objects and determine a. what type and degree of format conversion or other preservation actions should be applied b. the appropriate metadata needed for each object type and how it is associated with the objects 5. Estimate cost and lead time 6. Arrange for the necessary expertise to be available
Task C. Curation
Actions 7. Digitize data 8. Convert to a (CLARIN) preferred format 9. Assign appropriate metadata 10. Provide documentation
Task D. Validation
Actions 11. Validate curated resource
Task E. Archiving
Actions 12. Transfer to CLARIN Data Centre for long-term storage and maintenance 13. Assign persistent identifier 14. Provide access to content

Figure 1. Tasks and actions in data curation

Relevance to research community
As CLARIN-NL is directed at researchers in the humanities and social sciences, the infrastructure should incorporate the resources that are relevant to these research communities. Seeing that the field of Dutch language and speech technology is already very well organized and many resources are available through the HLT Agency, the curation of resources of interest to other areas is found to be relatively more urgent. Therefore in the Calls for proposals some priority areas have been identified solliciting project proposals targeted at literary studies, history and political studies, communication and media studies, first and second language acquisition, and historical linguistics.

Uniqueness
It may be argued that priority should be given to resources that are unique in their sort. To the extent that a resource bears resemblance to resources already available it should be established which are the characteristics that set it apart. Only then is there is basis for deciding whether it is interesting enough to be curated. What became already apparent with the initial inventory of potentially interesting resources is that some resources go under different names, while others that go under the same name are in fact different resources or at least different versions of a resource. An example is the Eindhoven corpus, which also goes by the name of Corpus Uit den Boogaart, and for which there appear to be different versions (e.g. a Meertens version and a Groningen version, while the HLT Agency distributes the Eindhoven Corpus VU version) without it being clear if, and if so how exactly, these versions differ.

Urgency
The urgency to curate a resource may arise for a variety of reasons. It may be that the people responsible for the resources are about to disappear or have already disappeared: Researchers who have completed their PhD research and moved elsewhere, people that have retired or are about to retire. With their departure the risk of data loss is very real. Even when the data can be traced successfully, the knowledge needed to curate them successfully (e.g. knowledge of the content, but also IPR-related matters) may be lacking. Another cause for urgency may be the limited life of magnetic and optical media and the fact that the software and devices needed to retrieve the recorded information are disappearing as they are being replaced. Finally, in the context of specific research certain resources are particularly welcomed as they fill remaining gaps.

Reproducibility of the resource
When considering the reproducibility of a resource, the first question to be addressed is whether the resource contains

- primary data, i.e. the original texts, images or recordings
- transcriptions, annotations and other forms of enrichment of the primary data
- derived data, e.g. a frequency list or a concordance

Primary data may be any of a wide range of materials, including data that were collected during field work, while conducting a survey among speakers of a particular language or dialect (incl. questionnaires and interviews), or while running an experiment in the laboratory (incl. stimuli), but also a corpus of texts, a grammar or a lexicon that has been compiled. Primary data cannot usually be reproduced, or if they can, reproduction requires an excessive effort. Primary data therefore have high priority.[6]

[6] Excepted are data sets that consist of data that have been collected more or less at random (i.e. without a priori formulated design criteria) from the internet and which

With transcriptions etc. a distinction should be made between enrichments that were obtained either manually/semi-automatically or as the result of a fully automatic process.
In the first case, recreating these enrichments will appear not be trivial, while at the same time it is unlikely that an identical result can be obtained. Such data should therefore be curated.
In the case of automatically produced enrichments, these on principle could be reproduced when required, assuming that the tool(s) that is/are needed to do so is/are indeed available.[7] A strong argument in favour of curating the enrichments nevertheless (i.e. even when the tools are available) is that the resource with the enrichments is readily usable, whereas users who are left to apply the tools themselves may find it beyond their capabilities to do so efficiently and/or successfully. Even users who do know how to handle the tools may appreciate not having to run complex and time-consuming processes.

Derived data are any kind of data that can be produced on the basis of (a subset of) a primary data set and/or its enrichtments. Derived data are not usually to be considered as a prime target for curation, as concordances, frequency lists and such can be generated on demand. There may, however, be occasions when the idea of curating derived data may be entertained and actually be given some follow up. This could be with derived data that come with a resource (e.g. the various frequency lists with the Spoken Dutch Corpus). It may well be that these data are particularly interesting in their own right for particular user groups (e.g. developers of teaching materials looking for a basic vocabulary list). Curation of derived data must also be considered for complex data sets where it is all but trivial to derive the data one is interested in (e.g. a list of the pronunciation variants of content words in Dutch as spoken by speakers originating from the Netherlands).

Wether a resource up for curation can indeed be successfully curated (Action A3) depends on:

The state of the resource
For any resource that is being considered for curation it must be established whether

- it can be made available to a wider audience; questions that need to be addressed here are: Has the resource been cleared for IPR?[8] Should measures be taken to ensure anonymization? Etc.
- it is in digital form; to the extent that resources are not in digital form, digitization is needed.

Other questions in this context are

- is the resource in a state and form that can still be handled by current hard- and software?
- is the integrity of the data as yet in tact?
- upon curation, can the integrity of the data be warranted
- is it in a sound state qualitatively?

The availability of documentation
Documentation may take on many different forms. It includes format specifications and descriptions, protocols, annotation guidelines, but also descriptions of the experimental design, the set-up and the stimuli used. The availability of proper (technical and user) documentation is one of the preconditions for curation to be successful, while it is also essential for ensuring that users can use the resource to the full.

The availability of expert knowledge
Expert knowledge of a scientist or the original collector may be indispensable when curation of a resource is to be undertaken and conversion of the original form to its projected form is not straightforward.

The availability of the necessary tools, scripts, etc.
To the extent that specific tools etc. are necessary for the curation of a resource, they should be available or it should be possible to develop them without disproportional effort.

After a candidate resource has been properly assessed, the next step in the curation process is to develop a curation plan. The plan should specify what actions are necessary to preserve the data and accompanying metadata. This may involve digitization and conversion to CLARIN preferred formats. From a very early stage in the curation process the designated CLARIN Data Centre that will eventually store and maintain the curated resource is involved. Elements of such a curation plan are addressed in more detail below.

4. The Case of the Dutch Bilingual Database, Roots of Ethnolects and TCULT collections

In this section we report on our experiences in the DCS with the curation of three data collections, viz. the Dutch Bilingual Database, Roots of Ethnolects and TCULT. They form an interesting and representative case for a number of reasons:

- the data over time have been produced and held at various locations, some data is presumed missing but chances are that these may yet be retrieved
- there are several types of data (audio recordings, transcripts, images and descriptions of materials used to elicit the data and protocols/descriptions of the task), metadata and formats (wav, mp3, mp4, jpg, mpeg, txt, pdf, chat, imdi) which –to the extent that

have no particularly distinctive characteristics. While exact reproduction may not be possible, it can assumed that similar data sets can be produced if so desired.

[7] The best strategy therefore would be to consider curating both the data and the tools. However, the curation of tools is complex and serious questions have been raised as to whether it is worth the effort.
[8] In case arrangements have yet to be made, a Creative Commons or similar licence is preferred.

they do not conform to one of the CLARIN preferred formats– should be converted, thus providing an ideal test case for applying available tools
- two CLARIN-NL data centres (the Meertens Institute and The Language Archive at the MPI) are involved as targeted archiving centres.

4.1 Description of the resources

The Dutch Bilingualism Database (DBD) is a rather substantial collection of data (over 1,500 sessions) from a number of projects and research programmes that were directed at investigating multilingualism and comprises data originating from Dutch, Sranan, Sarnami, Papiamentu, Arabic, Berber and Turkish speakers . At the basis of the collection is the research project TCULT[9] (1998-2002) in which intercultural language contacts in the Dutch city of Utrecht were studied. DBD established a first curation of the TCULT data and added many more bilingual data sets collected in the period 1985 – 2005. The current version of the DBD has IMDI[10] metadata files and is made accessible by the MPI at http://corpus1.mpi.nl/ds/imdi_browser/. The audio and text data are stored at the MPI and at the Meertens Institute. The DBD data consists of audio recordings, most of which are in WAV format while some are in MP3. Most transcripts that are available are in CHAT (Childes), some in TXT, PDF or EAF (Elan). Metadata are available in IMDI. In a number of cases additional metadata are available in TXT or PDF format. Occasionally, additional materials are available. These include descriptions (PDF) or images (JPG; PDF) of the pictures books or cartoons used to elicit the data and protocols/descriptions of the tasks involved (PDF).

Roots of Ethnolects is a well-structured collection of 168 audio recordings of Dutch, Arabic, Berber and Turkish. The data are stored at the Meertens Institute together with metadata (IMDI). For a number of recordings transcripts are available (EAF). In addition, protocols are available describing how the data were collected. Since this collection is well shaped and complete, the main efforts of curation at the DCS are directed towards the TCULT/DBD collections.

IPR

Permission for use of the DBD (incl. the TCULT) data was obtained from the subjects under the condition that they will be anonimized.[11] For the Roots of Ethnolects data subjects gave their consent and data may be used freely.

4.2 Development of a curation plan

We conducted interviews with the researchers involved in the TCULT/DBD collections. Based on what we learned from these sessions, we made an inventory of extra data that should be available, we made a list of metadata considered relevant and of preferred formats.

On the basis of our findings we will establish a curation plan (Action B in Figure 1). This plan addresses the following issues:

Restoring data:

Missing data are identified at two levels: A. missing files in currently available recordings (e.g .either audio files or transcription files); B. missing sessions that must/can be added to the collection. Whether data can be added to the curated corpus depends on the effort needed so as to make the material accessible (e.g. AD-conversion of audio, or scanning transcriptions available on paper).

Restructuring the corpus

The structure of the corpus in its present condition is very inconsistent. At the first level the data are divided according to language which is well justifiable. However, below this layer the structure becomes very diffuse. Subdirectories are introduced with names referring to a variety of metadata, e.g. collector, informant, city of recording. A more consistent approach is to put the session names as sublayer directly below the language directory since all the other information in subdirectory names is already part of the metadata of the recording session.

Converting the metadata to CMDI

Within CLARIN, CMDI[12] is the preferred format for metadata, IMDI can be considered as a predecessor format of CMDI. CMDI categories should be ISOcat[13] categories or be related to these (Windhouwer et al., 2010). For curation this involves a number of tasks:
- establish a list with relevant metadata categories for this corpus
- establish a mapping list showing in which IMDI fields these metadata occur, and, where appropriate, including additional metadata
- establish a mapping list of corresponding CMDI and ISOcat metadata categories

Upon curation of this resource the DCS simultaneously develops a CMDI profile for bilingual speech corpora which can be applied to other similar corpora. This implies that the profile includes metadata categories which are not present in the DBD, but are considered relevant for this type of corpus.

Converting data formats

The following conversion steps for DBD data have been identified:
- the conversion of MP3 to WAV
- the digitization of retrieved audio recordings (if so required)
- the conversion of DBD transcripts currently in TXT or PDF to CHAT or EAF format

[9] http://ebookbrowse.com/tcult-pdf-d68190469
[10] http://www.mpi.nl/isle/
[11] See the Conditions_for_use_DBD (under node DBD at http://corpus1.mpi.nl/ds/imdi_browser/).

[12] http://www.clarin.eu/cmdi
[13] http://www.isocat.org/

- converting the metadata (IMDI) to CMDI. Although the data are already stored at the MPI and the Meertens Institute and also the curation of ‘core’ IMDI to CMDI is available, the extensions to IMDI developed in the DBD project (the IMDI DBD profile) which were used for both the DBD and Roots of Ethnolects data remain to be converted.

It will be investigated to what extent it is feasible to convert all transcripts in CHAT to EAF format, using the tools provided by Brian McWhinney. The idea here would be that researchers working with both DBD and Roots of Ethnolects data could have all the data in a single format (EAF), if they chose to do so. Where transcripts exist in the CHAT format, both the CHAT and the EAF are to be maintained, which means that attention must be given also to the synchronization of these versions.

Once the individual tasks have been detailed, the corresponding personnel effort and costs must be estimated and included in the curation plan. After approval of the plan, the necessary personnel with the required expertise must be made available for the envisaged tasks. These activities need not per se be executed by DCS staff, but may be outsourced to third parties, such as CLARIN-NL Data Centres.

5. Conclusion

Researchers who possess valuable data that are on the verge of oblivion should be stimulated and guided to make these available and accessible to the research community and (where relevant) to the wider public. In this paper we have introduced the CLARIN-NL Data Curation Service that has been established for exactly this purpose. Of course the main task of the DCS is curating resources. However, before starting any curation the DCS has a clear obligation to assess the desirability and feasibility of the curation of a data resource. We have outlined the leading considerations underlying such a decision. Upon a positive decision, another relevant preparatory action is setting up a curation plan for the resource. We have illustrated this in our work on the DBD/TCULT database. Apart from the curation of this and other resources, an important task will be the identification and assessment of resources that qualify for curation in the near future.

Acknowledgement

The research for this paper was funded by CLARIN-NL (http://www.clarin.nl) under grant numbers CLARIN-NL-10-025 and CLARIN-NL-11-005.

References

Duranti, L. The long-term preservation of accurate and authentic digital data: the InterPARES project. In *Data Science Journal*, Volume 4, 25 October 2005: 106-118.

ELSNET’s HLT Roadmap. http://elsnet.dfki.de/ (November 2010).

Hedstrom, M. 1997. Digital preservation: a time bomb for Digital Libraries. In *Computers and the humanities*, 31(3): 189-202. Retrieved from http://www.uky.edu/~kiernan/DL/hedstrom.html.

Hedstrom, M., S. Ross, K. Ashley, B. Christensen-Dalsgaard, W. Duff, H. Gladney, C. Huc, A. Kenney, R. Moore & E. Neuhold. 2003. *Invest to Save. Report and recommendations of the NSF-DELOS working Group on digital archiving and preservation*. Prepared for National Science Foundation (NSF) Digital Library Initiative & The European Union under the Fifth Framework Programme by the Network of Excellence for Digital Libraries (DELOS). Retrieved from http://delos-noe.iei.pi.cnr.it/ activities/ internationalforum/Joint-WGs/digitalarchiving/ Digitalarchiving.pdf

Gray, J., A. Szalay, A. Thakar, C. Stroughton, J. vanden Berg. 2002. *Online Scientific Data Curation, Publication and Archiving*. Technical Report MSR-TR-2002-74. Redmond, Microsoft Research. Retrieved from http://research.microsoft.com/apps/pubs/default.aspx?id=64568.

Nauta, G.-J., R. Grim, I. Angevaare, H. Tjalsma, A. van Nispen & A. van der Kuil (eds.). 2010. *Data curation in arts and media research*. Stichting SURF. Retrieved from http://www.surffoundation.nl/nl/publicaties/Pages/StudieDataCurationinArtsandMediaResearch.aspx.

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

Oostdijk, N. 2011. *CLARIN-NL Data Curation Service*. CLARIN-NL internal publication. Retrievable from …

Russell, K. (ed.). 2010. *IISH Guidelines for preserving research data. A framework for preserving collaborative data collections for future research*. Stichting SURF. Retrieved from http://www.surffoundation.nl/nl/publicaties/Pages/StudieIISHGuidelinesforpreservingresearchdata.aspx.

Tjalsma, H. & A. van der Kuil (eds.). 2010. *Selection of research data. Guidelines for appraising and selecting research data.* A report by DANS and 3TU.Datacentrum. Stichting SURF. Retrieved from http://www.surffoundation.nl/nl/publicaties/Pages/StudieSelectionofResearchData.aspx.

Trusted digital repositories: Attributes and responsibilities. An RLG-OCLC report. 2002. RLG, Mountain View, CA. Retrieved from http://www.oclc.org/research/activities/past/rlg/trustedrep/repositories.pdf.

Windhouwer M, Wright SE, Kemps-Snijders M (2010) *Referencing ISOcat data categories*. In Budin G, Declerck T, Romary L, Wittenburg P (eds) Proceedings of the LREC 2010 LRT standards workshop, Malta, http://www.lrecconf.org/proceedings/lrec2010/workshops/W4.pdf

Evaluating DBMS-based Access Strategies to Very Large Multi-layer Corpora

Roman Schneider
Institut für deutsche Sprache (IDS)
R5 6-13, D-68161 Mannheim /Germany
schneider@ids-mannheim.de

Abstract

Linguistic query systems are special purpose IR applications. As text sizes, annotation layers, and metadata schemes of language corpora grow rapidly, performing complex searches becomes a highly computational expensive task. We evaluate several storage models and indexing variants in two multi-processor/multi-core environments, focusing on prototypical linguistic querying scenarios. Our aim is to reveal modeling and querying tendencies – rather than absolute benchmark results – when using a relational database management system (RDBMS) and MapReduce for natural language corpus retrieval. Based on these findings, we are going to improve our approach for the efficient exploitation of very large corpora, combining advantages of state-of-the-art database systems with decomposition/parallelization strategies. Our reference implementation uses the German DeReKo reference corpus with currently more than 4 billion word forms, various multi-layer linguistic annotations, and several types of text-specific metadata. The proposed strategy is language-independent and adaptable to large-scale multilingual corpora.

Keywords: Very Large Corpora, Multi-layer Annotation, Linguistic Retrieval, Database Management Systems, Concurrency

1. Motivation

In recent years, the quantitative examination of natural language phenomena has become one of the predominant paradigms within (computational) linguistics. Both fundamental research on the basic principles of human language as well as the development of speech and language technology increasingly rely on the empirical verification of assumptions, rules, and theories. More data are mostly seen as better data (Church & Mercer, 1993), and consequently, we notice a growing number of national and international initiatives related to the building of large linguistic datasets for contemporary world languages. Besides written (and sometimes spoken) language samples, these corpora usually contain vast collections of morphosyntactic, phonetic, semantic, etc. annotations, plus text- or corpus-specific metadata.

The downside of this trend is obvious: Complex queries against very large multi-layer corpora (meaning corpora with multiple, potentially concurring annotation layers) quickly become highly computational expensive tasks. So even with specialized applications, our ability to store linguistic data is often bigger than our ability to analyze all this data in detail. In addition, the findings of empirical corpus studies should be traceable and reproducible (Kilgarriff, 2007; Pedersen, 2008).

Much of essential work towards the querying of linguistic corpora goes into data representation, integration of different annotation systems, and the formulation of query languages (Rehm et. al., 2008; Zeldes et. al., 2009; Kepser et. al., 2010; Frick et. al., 2012). We add to this efforts by focusing the scaling problem: As we go beyond corpus sizes of some billion words and at the same time increase the number of possible search keys (linguistically motivated annotations as well as text-specific metadata like publication date, text type, genre, etc.), query costs rise disproportionately. This is due to the fact that unlike traditional IR systems, corpus retrieval systems have to deal not only with the "horizontal" representation of textual data but also with heterogeneous metadata on all levels of linguistic description. And, of course, the exploration of inter-relationships between annotations becomes more and more challenging as the number of annotation systems increases; e.g., Bański et. al. (2012) discuss the demands and technical issues when developing innovative corpus analysis platforms.

Given this context, we proposed a novel retrieval approach that uses task parallelization and scales well to billion-word corpora (Schneider, 2011). The following sections evaluate speedup effects of this approach in multi-processor/multi-core environments as well as influences of data parallelization (partitioning) and indexing methods.

2. The Reference System

We are using a subset of 4 billion words from the multi-layer annotated German Reference Corpus *DeReKo (Deutsches Referenzkorpus)* (Kupietz et. al., 2010), which constitutes the largest linguistically motivated collection of contemporary German. It covers language data from different media types (literature, newspapers, specialist journals, online texts, etc.), has been annotated morphosyntactically with three competing systems (Connexor, Xerox, TreeTagger), and provides additional text-specific metadata.

Complex data of this kind can be made accessible in different ways. In order to filter out inappropriate approaches, we formulate the following presuppositions:

i. XML/SGML-based markup languages are more suitable for data exchange than for efficient storing and retrieval, so we prefer a compact encoding.

ii. File-based data storage is less robust, flexible, and powerful than the maintenance of text and metadata within database management systems (DBMS).

iii. Although a computer's main memory is still the fastest form of data storage, even with compression techniques it does not seam feasible to rely exclusively on RAM. Attempts to implement (indexless) in-memory databases for considerably large language corpora (Pomikálek et. al., 2009) perform well for unparsed texts but are strongly limited in terms of storage size and, therefore, cannot deal with terabytes of multi-layer annotations.

iv. In order to overcome physical RAM limitations, other approaches use database systems and decompose sequences of strings from the source texts into indexed n-gram tables (Davies, 2005). This results in relatively low query costs and allows fast retrieval with a predefined maximum number of search expressions. However, space requirements for increasing values of n are enormous, and the consuming of system resources for queries spanning more than a handful of words or even sentences, thus, becomes unacceptable. Moreover, complex queries with regular expressions (*Find all tokens that start with a capital letter followed by one or two vocals, and end on 'der'*) or NOT-queries (*Find all tokens that are not classified as nouns*) – both are crucial for comprehensive linguistic exploration – do not benefit from n-gram-based full-string indexes and, thus, perform rather poor.

As a consequence, our corpus storage and retrieval approach uses an object-relational DBMS (64-bit Oracle 11.2.0.1.0 running on CentOS 5.6) with fine-grained data spread across specifically designed tables. Figure 1 shows an excerpt from our conceptual data model as an entity-relationship diagram. Entity types (corpus, text, sentence, word) are displayed as rectangles with their attributes represented as ellipses (computed attributes have dotted borders); relationships are represented as diamond-shape connectors. In order to evaluate speedup and scaling effects, we implemented the entire framework (i.e., data and retrieval procedures) on two independent systems:

i. Single computer, single processor, multi-core: A commodity low-end server driven by a quad-core CPU with 2.67 GHz clock rate and 16GB RAM.

ii. Single computer, multi-processor, multi-core: This symmetric multiprocessing system (SMP) uses eight quad-core microprocessors with 2,3 GHz clock rate and 128GB RAM.

For the reliable measurement of query execution times – and especially to minimize caching effects – we used a "cold" database, meaning that the database instance is opened and that the most relevant caching areas are cleared. Oracle's server-side memory, known as System Global Area (SGA), consists of various components dedicated to different tasks: The dictionary cache holds information about data dictionary objects; the redo log buffer stores uncommitted transactions etc. Most influential for our retrieval runs is probably the buffer cache, i.e., the part of the SGA containing the most recently used data blocks in order to reduce disk I/O. We always cleared it by entering the command ALTER SYSTEM FLUSH BUFFER_CACHE. We intentionally did not flush the Shared Pool (ALTER SYSTEM FLUSH SHARED_POOL) because this part of the SGA stores, among other things, information about user privileges, table structures, etc. as well as optimized execution plans. We believe that especially the latter are essential features of a relational database system and should be taken into account when evaluating its suitability for corpus retrieval - flushing the shared pool seems even more artificial than not flushing it.

3. Query Strategies

Focusing on DBMS-driven corpus storage, the following retrieval strategies look promising:

i. Concatenated SQL joins: This strategy makes use of the relational power of sub-queries and joins and is already used for productive corpus retrieval. Chiarcos et. al. (2008) use an intermediate language between query formulation and database backend; Bird et. al. (2005) present an algorithm for the direct translation of linguistic queries into SQL. This approach uses absolute word positions, and therefore allows proximity queries without limitation of word distances. We implemented an extensible web-based retrieval form (see Figure 2) that can be used to intuitively formulate complex queries using distinct search keys on different metadata types, e.g.: *Find all sentences containing a determiner immediately followed by a proper noun ending on "er", immediately followed by a noun, immediately followed by the lemma "oder", followed by a determiner (any distance), immediately followed by a plural noun, followed by the lemma "sein" (any distance)*. This form invokes a stored procedure that formulates and executes the corresponding SQL query with all necessary joins. The query uses one single token table, corresponding to the "word" entity in Figure 1.

ii. Task separation and parallelization: Programming models like MapReduce support concurrent execution of tasks and, thus, tackle large-data problems. Alhough MapReduce is already in use in a wide range of data-intensive applications (Lin & Dyer, 2010), its principle of "divide and conquer" has not yet been employed for corpus retrieval. We employ an extended MapReduce strategy that regulates the distribution of language data and processor-intensive computation over several CPU cores – or even cluster of machines – and controls the partition of complex linguistic queries into independent processes that can be executed in parallel. Figure 3 illustrates the map/reduce processes for our sample query from above: Within a "map" step, the original query is partitioned into eight separate key-value pairs. Keys represent linguistic units (position, token, lemma, part-of-speech, etc.) values may be the actual content. Again, all queries use the single word table, but they can be processed in parallel and pass their results (sentence/position) to temporary tables. The subsequent "reduce" processes filter out inappropriate results step by step. Usually, multiple reducing steps cannot be executed in parallel because each reduction produces the basis for the next step. But our framework, implemented with the help of PL/SQL stored procedures within the RDBMS, overcomes this restriction by dividing the entire search tree into multiple sub-trees. The reduce processes for different sub-trees are scheduled simultaneously and aggregate their results after they are finished. So the seven reduce steps of our example can be executed quite naturally within only four parallel stages. The parallel framework stores the search results within the database schema, making it easy to reuse them for further statistical processing. Additional metadata restrictions (e.g., genre, topic, location, date) are translated into separate map processes and reduced/merged in parallel to the main search.

4. Evaluation Scenarios and Results

Executing the concatenated SQL statement (3.i) with eight multi-type search keys against the four-billion word corpus on the single processor system exceeded its capability when applying the joins with traditional B-tree indexes on unpartitioned heap-tables because the nested loops generated an immense workload – we canceled the operation after a runtime of ten hours. The parallel MapReduce search (3.ii) took less than thirty minutes to complete. This strongly indicates that the second approach fits much better for big corpus data and multiple search keys but that further improvements should be carried out. This includes the testing of appropriate index variants that are the key to efficient corpus retrieval (Ghodke & Bird, 2010).

Although we are comparing response times for queries on different server systems and under different settings, our main interest is not to contribute to overall benchmark tests. Database management systems are a widespread and mature technology: Their general advantages and disadvantages have been listed and benchmarked for decades. The most prominent feature that makes them interesting for querying multiple annotated language corpora is probably the flexibility of the (object-)relational approach: Multiple markup layers can be converted into object-relational structures that then can be accessed with the help of SQL; additional metadata such as text type or creation date can be added and queried in the same way. But despite these general advantages, the practical application of database management systems for corpus retrieval is still under-investigated. We aim to reveal tendencies when using RDBMSs and MapReduce for natural language corpus retrieval: Which query strategy seems reasonable under certain conditions? Which storing settings fit to specific language data? Thus, our systematic evaluation concentrates on the following questions:

i. How do SQL joins perform for increasing numbers of search keys? We evaluate this on 1-, 10-, 100-, 1000-, and 4000-million word corpora with rare-, low-, mid-, high-, and top-level search keys. Figure 4 shows the response times in seconds on the single processor server for the query *select count(t1.co_sentenceid) from tb_token t1, (select co_id, co_sentenceid from tb_token where co_token=token1) t3, (select co_id, co_sentenceid from tb_token where co_token = token2) t2 where co_token = token3 and t1.co_sentenceid = t2.co_sentenceid and t1.co_sentenceid = t3.co_sentenceid and t1.co_id < t2.co_id and t2.co_id < t3.co_id*, using three search keys on identical annotation data (token) and single-column indexes that can be queried in parallel. This query simply counts the number of sentences containing three specified tokens (token1, token2, token3) in a fixed order. Compared to similar queries with two search keys (63s for a top-level search, Figure 5) or one search key (6s for a top-level search, Figure 6), the increase of response time on the 4000-million corpus is obviously disproportional (301s). So SQL joins on token data get remarkably less performant for searches with three (or even more) top-frequent search keys, even when making use of in-built query parallelization and the database's cost-based optimizer. On the multi-processor server, the results showed the same tendency: Alhough absolute response times were decreased by a factor of 5 to 6 (e.g., the search for *der* and *die* on the 4000-million corpus completed in 14 seconds instead of 63 seconds, the search for *der*, *die* and *und* completed in 56 seconds instead of 301 seconds), queries still took significantly longer when joining three (or even more) frequent search tokens in a SQL statement.

ii. When sticking to a maximum of two search keys per SQL query (and splitting queries with more search keys to multiple “map” processes that store their results in temporary tables), how does this solution scale on the multi-processor system? We tested this for three top-frequent tokens (*der/die/und*) on the 4000-million corpus. The search pattern is – from a linguistic point of view – not genuinely interesting, neither semantically nor grammatically. But it provides a perfect test-case for "expensive" corpus requests by initiating the following highly data-intensive SQL statements:

(1) MAP1: *insert into TB_MAP1 (co_sentenceid, co_id) select t1.co_sentenceid, t2.co_id from TB_TOKEN t1, (select co_id, co_sentenceid from TB_TOKEN where co_token='die') t2 where t1.co_token='der' and t1.co_sentenceid = t2.co_sentenceid and t1.co_id < t2.co_id;*

(2) MAP2: *insert into TB_MAP2 (co_sentenceid, co_id) select t1.co_sentenceid, t1.co_id from TB_TOKEN t1 where t1.co_token='und';*

(3) REDUCE: *insert into TB_REDUCE (co_sentenceid) select t1.co_sentenceid from TB_MAP1 t1, TB_MAP2 t2 where t1.co_sentenceid=t2.co_sentenceid and t1.co_id < t2.co_id;*

Statement 1 and statement 2 are scheduled simultaneously; the reduce statement 3 has to wait until both map processes are completed. Figure 7 shows the results on the two servers – please note that – as expected – the insert statements took longer than the queries in 4.i since we are not only counting the number of sentences but also storing the sentence ids in a final result table for further processing. Overall, our tests reveal that the scaling benefit is not strictly linear (32 CPU cores do not perform our MapReduce retrieval eight times faster than 4 cores), but it is promising: The searches on the multi-processor system completed about 4 to 5 times faster than on the single processor system. This corresponds to Amdahl's Law that says that the maximum speedup improvement when using multiple processors is limited by several factors, most prominently by inevitable sequential fractions of the executed tasks. On the other hand, concurrency problems gain weight when parallelization is increased; this phenomenon is addressed by Neil Gunther's Super-Serial Scalability Model. At any rate, our MapReduce approach performed better than the concatenated SQL joins – and the more search keys are used, the difference should be bigger. There may be some potential for further optimization of our mapping algorithm (How should complex queries be divided? How many parallel query tasks are optimal for our system?) as well as for the fine-tuning of database parameters (sizes of memory areas, maximum number of processes, parallelization degrees of tables and indexes, etc.).

iii. How does partitioning of language data improve response times? We partitioned a separate POS table, containing information about each token's word class as identified by the Connexor tagger, according to POS value and sentence number: *create table tb_morpho (co_sentenceid number(10), co_morpho varchar2(10), co_sub varchar2(10), co_id number(10)) partition by range(co_sentenceid) subpartition by list (co_morpho) subpartition template (subpartition A values ('A'), subpartition DET values ('DET'), ... , subpartition PRON values ('PRON'), subpartition P values ('P')) (partition p1 values less than (20000000), partition p2 values less than (40000000), ... , partition p11 values less than (220000000), partition p12 values less than (240000000), partition p13 values less than (MAXVALUE))*. The same partitioning was done for the associated multi-column index. When comparing response times after and before the reorganization, we found that the query s*elect unique t1.co_sentenceid from* *tb_morpho t1, tb_morpho t2 where t1.co_sentenceid = t2.co_sentenceid and t1.co_morpho = 'PRON' and t2.co_morpho='DET'* ("find all sentences containing a pronoun and a determiner") on the single processor system was now completed within 50 seconds instead of 300 seconds. So partitioning relational tables holding linguistic data according to often-used search attributes with a small number of distinct values obviously raises their potential for fast query execution. This can be explained with the simple fact that queries for particular POS values no longer have to scan the whole table/index but only certain partitions, and with the possibility to distribute searches on multiple partitions to multiple CPU cores.

iv. How can specific index types improve complex queries? Advanced pattern matching with regular expressions (RegExp) or double/left truncated wildcards is a feature often demanded for linguistic corpus retrieval. Modern database management systems usually contain specific enhancements for their retrieval languages that allow for the integration of regular expressions. For example, Oracle RDBMS offers four RegExp functions that implement the POSIX Extended Regular Expressions (ERE): REGEXP_REPLACE, REGEXP_SUBSTR, REGEXP_LIKE and REGEXP_INSTR. E.g., a SQL search for tokens starting with a capital letter, followed by the substring *'mini'* (at any distance) and ending on the suffix '*er*' (again at any distance) would use the RegExp-enhanced restriction clause *WHERE REGEXP_LIKE (<column name>, '^[[:upper:]].*mini.*er$')*. Unfortunately, standard database index types (b-tree, bitmap) do not support this kind of query because it is impossible to forecast all possible RegExp patterns, so the execution plan will always arrange a time-consuming full table scan. On the other hand, prototypical linguistic searches mostly contain indexable substrings that can be used to speed up the query: Linguists look quite rarely for complex chemical formulas or something like "*a word starting with two numeric digits, followed by 'A' or 'D', followed by a numeric digit between 5 and 8, etc*.". More frequently, they are interested in words ending on a certain suffix or containing a certain stem. In order to improve performance for such queries, Giles (2005) propose to build functional bitmap indexes over all possible substrings of all corpus tokens. These indexes should then be used as primary filters for queries containing regular expressions. Thus, the above query would be superseded by the statement *SELECT unique matchvalue FROM TABLE (getmatches ('tb_token', 'co_token', 'mini')) WHERE REGEXP_LIKE (matchvalue, '^[[:upper:]].*mini.*er$')*, where the in-line table function acts as primary filter by generating a subquery like *select /*+ index_combine(t) */ co_token from tb_token t where (substr(t.co_token, 1,2) = 'mi' and substr(t.co_token, 3,2) = 'ni') or (substr(t.co_token, 2,2) = 'mi' and substr(t.co_token, 4,2) = 'ni') [...]*. Since bitmap indexes can be combined efficiently on the fly, the subquery is expected to complete quite performant. We evaluated this approach and contrasted the results with:

(1) a similar approach that instead of user-built substring indexes uses Oracle's CONTEXT index type (Oracle Corp., 2011). CONTEXT is often used for building text query applications and document classification applications, and provides means to improve left-truncated and double-truncated wildcard searches.

(2) the omission of any primary filter, i.e. the full table scan variant.

Figure 8 addresses the substring filtering and shows the results and execution times (hh:mm:ss) of three sample SQL statements (without RegExp, but with substring search) that insert the retrieved unique tokens into a temporary table, using the two different primary filters on our single processor server. There is a minor difference regarding the number of results that can be explained by the fact that the manually-built substring indexes cover only the first 100 characters of each token (the corpus contains some outliers with up to 800 characters that we did not index), whereas CONTEXT includes all tokens. Both variants use the four CPU cores in parallel, but the CONTEXT index beats the user-built substring indexes clearly in terms of speed.

Next, we excluded the table inserts as well as the "unique" constraints, and tested some assorted RegExp-enhanced SQL statements that count the occurrences of all matching tokens within our corpus database. The first retrieval run was always without primary filters, then we used the manually-built substring bitmap indexes, and finally the built-in CONTEXT index. Figure 9 displays the corresponding results and response times. They reveal a somewhat unexpected behaviour, since the (indexless) full table scan variant sometimes performed considerably faster than queries with the user-built substring prefiltering. This behaviour did not – at least not always and not significantly – correspond with the amount of data/number of rows that were prefiltered/retrieved. Most likely, several issues generally influence the suitability of the tested approaches for regular expression search: The complexity of the RegExp search pattern, the use of left- or right-side truncation, the language-specific distribution of (initial) word substrings, etc. All these aspects are worth to be investigated in more detail; we see our test runs as a first starting point, outlining the way to an optimal use of database indexing techniques for RegExp retrieval. In any case, however, the best results were again achieved when we deployed the prefiltering restriction with the CONTEXT index. As we see in Figures 10 (full table scan) and 11 (CONTEXT index), the database's explain plan – automatically generated by the Cost Based Optimizer (CBO) with the help of table/index statistics – for this scenario starts with scanning the index, secondly applies the regular expression on the results, and only then accesses the table ("by local index rowid", the most cost-intensive operation of this run).

5. Summary and Outlook

The results of our studies demonstrate that the joining of relational DBMS technology with a parallel computing approach like MapReduce combines the best of both worlds for linguistically motivated corpus retrieval on big datasets. It makes annotated language corpora manageable, eases the reuse and further processing of results, and scales well. As our initial evaluation shows, standard SQL joins do not sufficiently handle queries for complex structures and syntagmatic patterns on very large natural language collections and should be complemented by a dedicated concurrency model. The tests on the multi-processor system demonstrate the suitability of our approach on high-end servers – and further parallelization over multiple machines would most likely benefit even more from the separation of sub-tasks/sub-queries.

Furthermore, we showed that in order to overcome relational bottlenecks, advanced database features like table partitioning and functional indexes can be adapted to the specifics of language corpora. For example, the value-driven partitioning of the POS table significantly improved response times, and introducing custom-tailored substring indexes for the pre-filtering of regular expressions seems to be an effective way to enable advanced pattern matching on big textual data. We will investigate the latter idea in more detail. In addition to the presented retrieval runs, a quick test with flexible combinations of manually-built indexes for word beginnings/endings (to be automatically employed for search expressions with fixed beginnings/endings) and

CONTEXT indexes (for double-truncated RegExp searches) already showed promising results.

For the future, we plan some scheduling refinements of our parallel framework as well as further testing of index types for different types of linguistic data. A comparison with existing corpus retrieval systems (e.g., *Corpus Workbench* or *SketchEngine*) and/or an implementation of our framework on different database management systems would be desirable – not only in terms of response times but also especially regarding flexibility (e.g., how can query results be displayed/processed?) and expandability (e.g., how can additional metadata and concurrent annotation layers be integrated?). In addition, it would be interesting to see how quantitative language laws can be utilized to fine-tune corpus retrieval systems, e.g., how Zipf's law can contribute to the design and filling of token tables.

6. References

Bański, P.; Fischer, P.M.; Frick, E.; Ketzan, E.; Kupietz, M.; Schnober, C.; Schonefeld, O.; Witt, A. (2012). The New IDS Corpus Analysis Platform: Challenges and Prospects. In Proceedings of the Eighth Conference on Language Resources and Evaluation.

Bird, S.; Chen, Y.; Davidson, S.; Lee, H.; Zhen, Y. (2005). Extending XPath to Support Linguistic Queries. In Proceedings of the Workshop on Programming Language Technologies for XML.

Chiarcos, C.; Dipper, S.; Götze, M.; Leser, U.; Lüdeling, A.; Ritz, J.; Stede, M. (2008). A Flexible Framework for Integrating Annotations from Different Tools and Tag Sets. In Traitement Automatique des Langues 49(2), pp. 271-293.

Church, K.; Mercer, R. (1993). Introduction to the Special Issue on Computational Linguistics Using Large Corpora. In Computational Linguistics 19(1), pp. 1-24.

Davies, M. (2005). The advantage of using relational databases for large corpora. In International Journal of Corpus Linguistics 10(3), pp. 307-334.

Frick, E.; Schnober, C.; Bański, P. (2012). Evaluating query languages for a corpus processing system. In Proceedings of the Eighth International Conference on Language Resources and Evaluation.

Giles, D. (2005). Oracle bitmap Indexes and their use in pattern matching. Unpublished Technical Whitepaper. *http://dominicgiles.com/technicalpapers.html*

Ghodke, S.; Bird, S. (2010). Fast query for large treebanks. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 267-275.

Kepser, S.; Mönnich, U.; Morawietz, F. (2010). Regular Query Techniques for XML-Documents. In Metzing, D.; Witt, A. (Eds.): Linguistic Modeling of Information and Markup Languages, Springer, pp. 249-266.

Kilgarriff, A. (2007). Googleology is bad science. In Computational Linguistics, 33(1), pp. 147-151.

Kupietz, M.; Belica, C.; Keibel, H.; Witt, A. (2010). The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In Proceedings of the Seventh Conference on Language Resources and Evaluation (LREC 2010), pp. 1848-1854.

Lin, J.; Dyer, C. (2010). Data-Intensive Text Processing with MapReduce. Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Oracle Corp. (2011). Text Application Developer's Guide 11*g* Release 2 (11.2). *http://docs.oracle.com/cd/E11882_01/text.112/e24435.pdf*

Pedersen, T. (2008). Empiricism Is Not a Matter of Faith. In Computational Linguistics, 34(3), pp. 465-470.

Pomikálek, J.; Rychlý, P.; Kilgarriff, A. (2009). Scaling to Billion-plus Word Corpora. In Advances in Computational Linguistics 41, pp. 3-13.

Rehm, G.; Schonefeld, O.; Witt, A.; Chiarcos, C.; Lehmberg, T. (2008). A Web-Platform for Preserving, Exploring, Visualising and Querying Linguistic Corpora and Other Resources. In Procesamiento del Lenguaje Natural 41, pp. 155-162.

Schneider, R. (2011). A functional database framework for querying very large multi-layer corpora. In Proceedings of the GSCL Conference 2011.

Zeldes, A.; Ritz, J.; Lüdeling, A.; Chiarcos, C. (2009). ANNIS: A Search Tool for Multi-Layer Annotated Corpora. In Proceedings of Corpus Linguistics 2009.

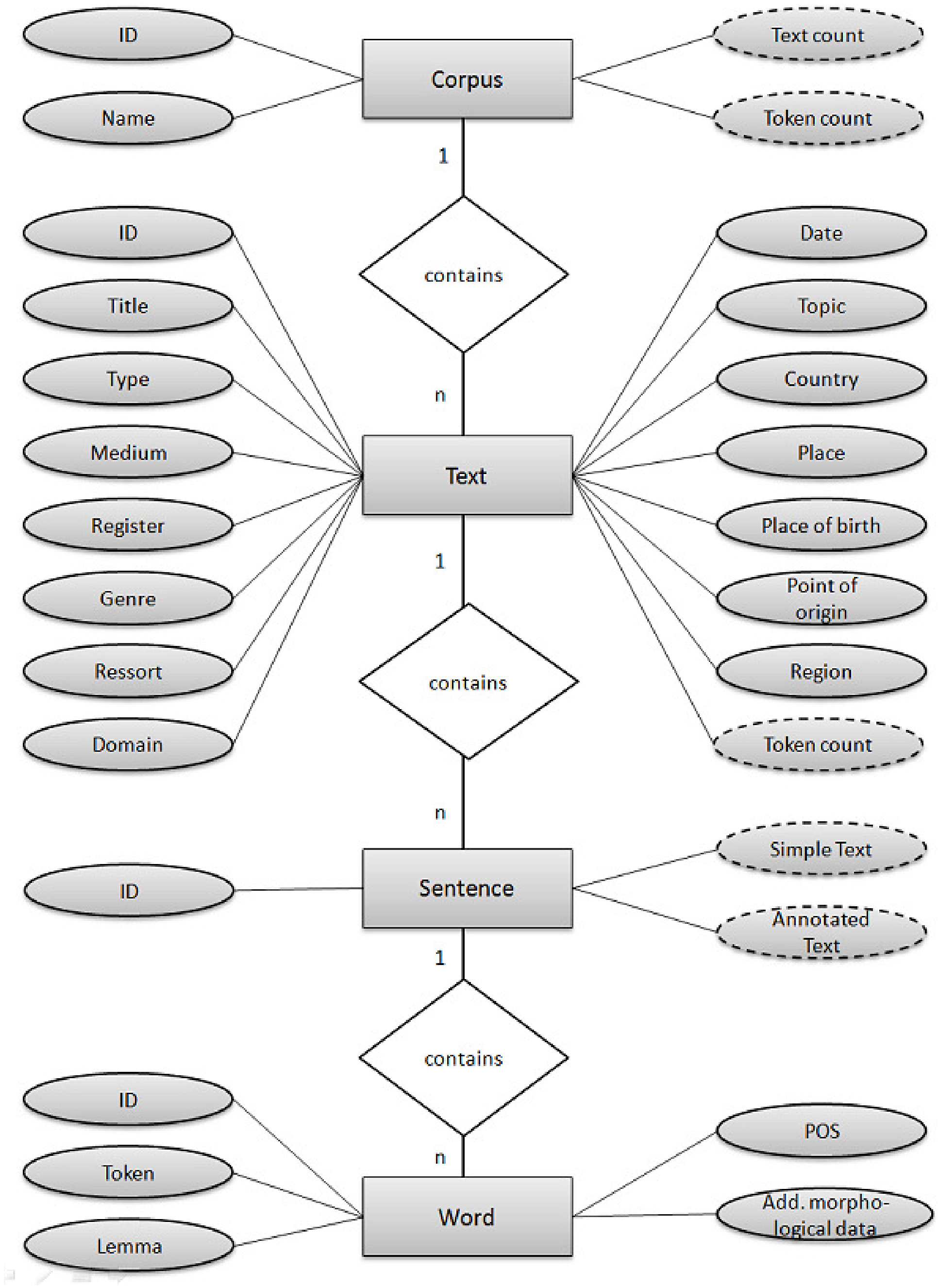
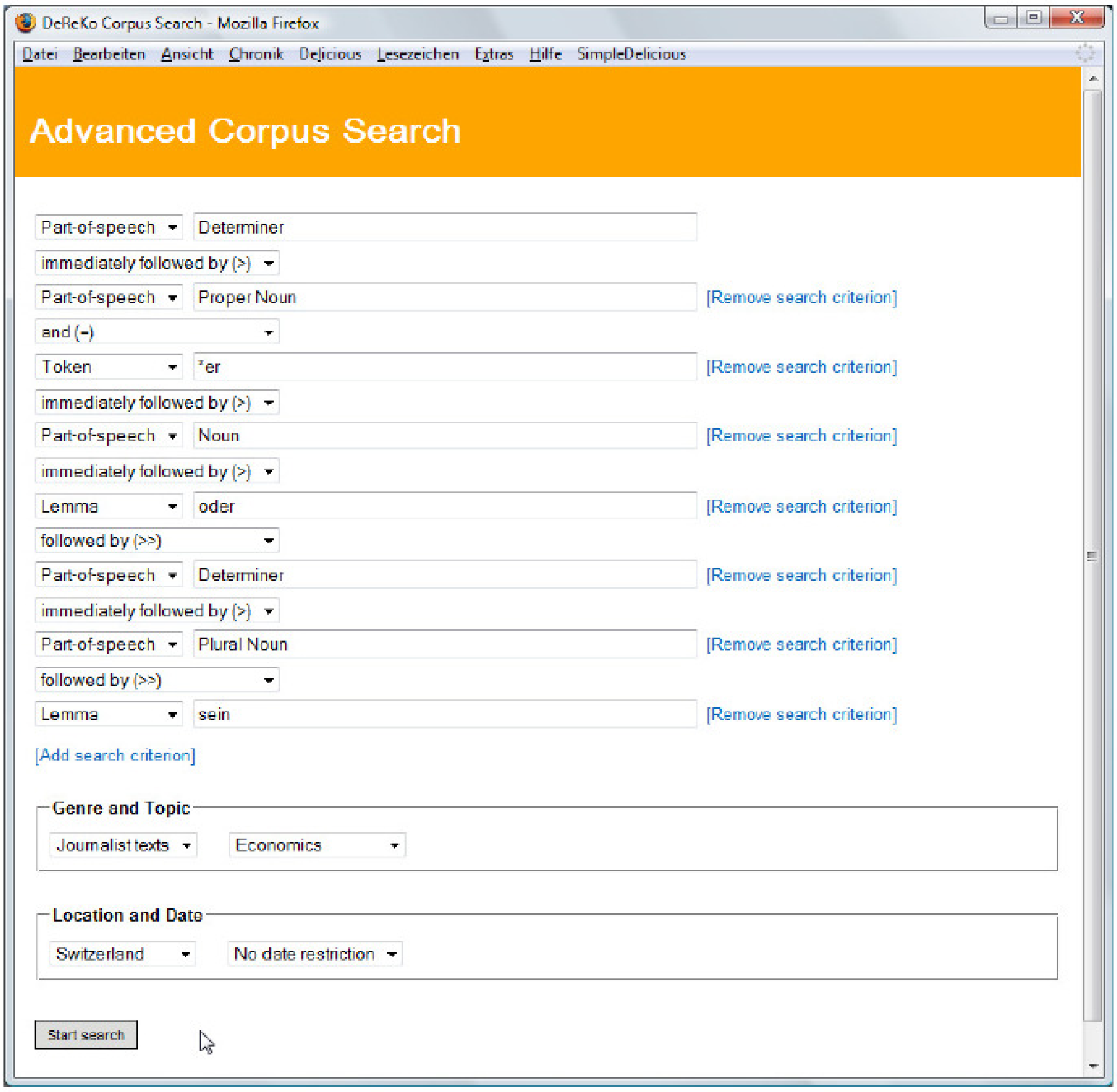


Figure 1: Semantic/conceptual data model (excerpt).


DeReKo Corpus Search - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Delicious Lesezeichen Extras Hilfe SimpleDelicious

Advanced Corpus Search

Part-of-speech	Determiner	
immediately followed by (>)		
Part-of-speech	Proper Noun	[Remove search criterion]
and (=)		
Token	*er	[Remove search criterion]
immediately followed by (>)		
Part-of-speech	Noun	[Remove search criterion]
immediately followed by (>)		
Lemma	oder	[Remove search criterion]
followed by (>>)		
Part-of-speech	Determiner	[Remove search criterion]
immediately followed by (>)		
Part-of-speech	Plural Noun	[Remove search criterion]
followed by (>>)		
Lemma	sein	[Remove search criterion]

[Add search criterion]

Genre and Topic

Journalist texts | Economics

Location and Date

Switzerland | No date restriction

Start search

Figure 2: Web-based retrieval form with our sample query.

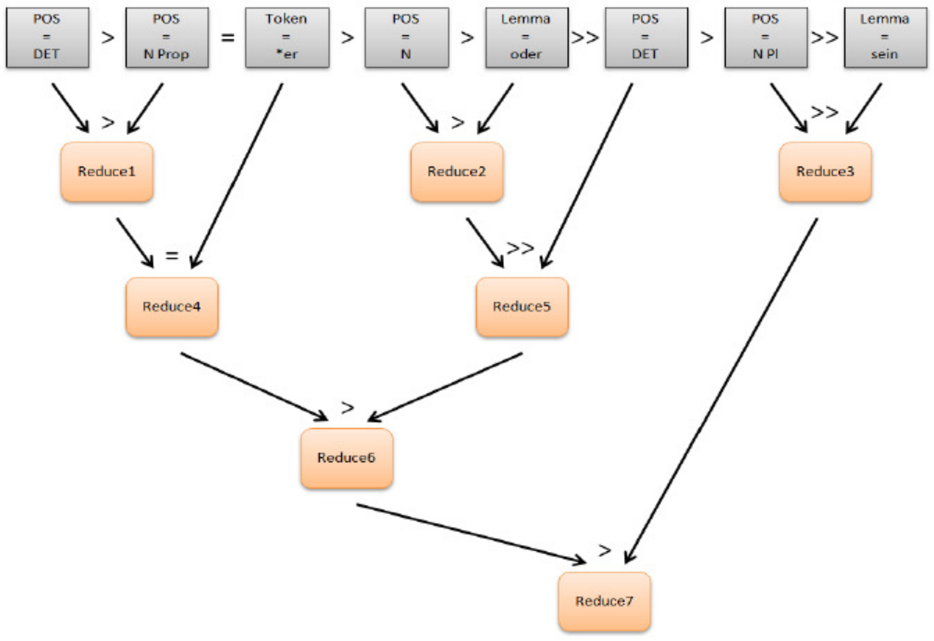


Figure 3: MapReduce processes for a concatenated query with eight search keys.

	1 Mio	10 Mio	100 Mio	1000 Mio	4000 Mio
rare (*traurige/isoliert/trauen*)	0,03	0,16	0,3	0,6	0,9
low (*langsam/lesen/verfügt*)	0,29	0,35	0,37	0,65	1
mid (*ohne/nun/uns*)	0,3	0,4	0,5	1,2	3,8
high (*nicht/ist/dem*)	0,31	0,47	1,49	10,47	38,7
top (*der/die/und*)	0,4	0,87	4,67	47,06	301

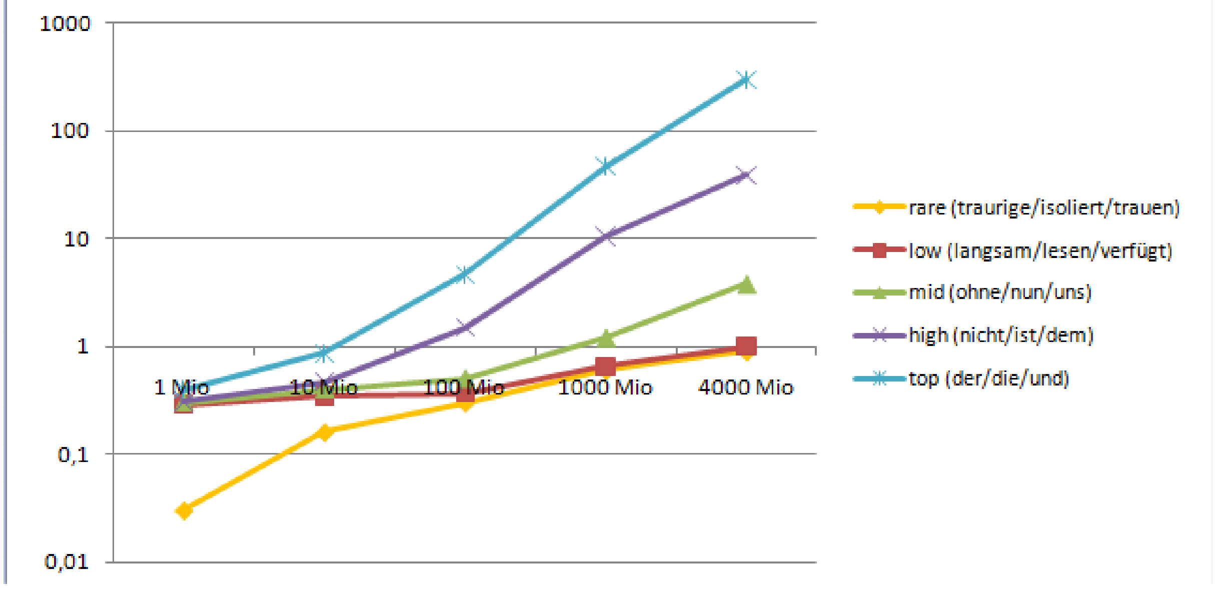


Figure 4: Response times (s) for nested SQL queries with three search keys (logarithmic scaled axis).

	1 Mio	10 Mio	100 Mio	1000 Mio	4000 Mio
rare (*traurige/isoliert*)	0,16	0,19	0,29	0,3	0,5
low (*langsam/lesen*)	0,09	0,2	0,3	0,4	1,3
mid (*ohne/nun*)	0,16	0,23	0,31	0,9	2,3
high (*nicht/ist*)	0,21	0,32	0,89	6,8	12,7
top (*der/die*)	0,32	0,65	3,12	56	63,3

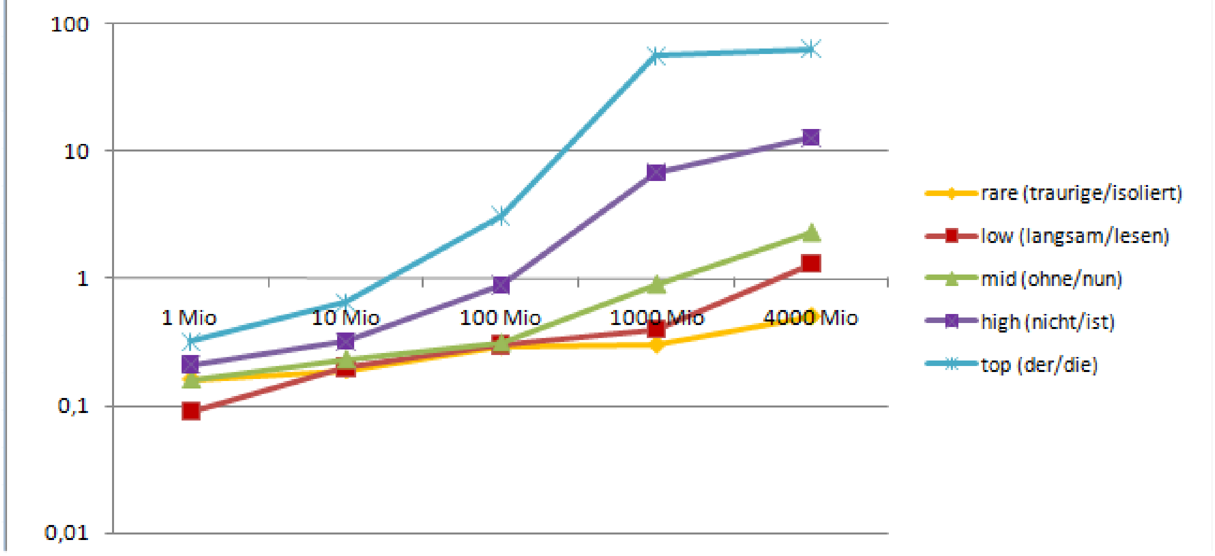
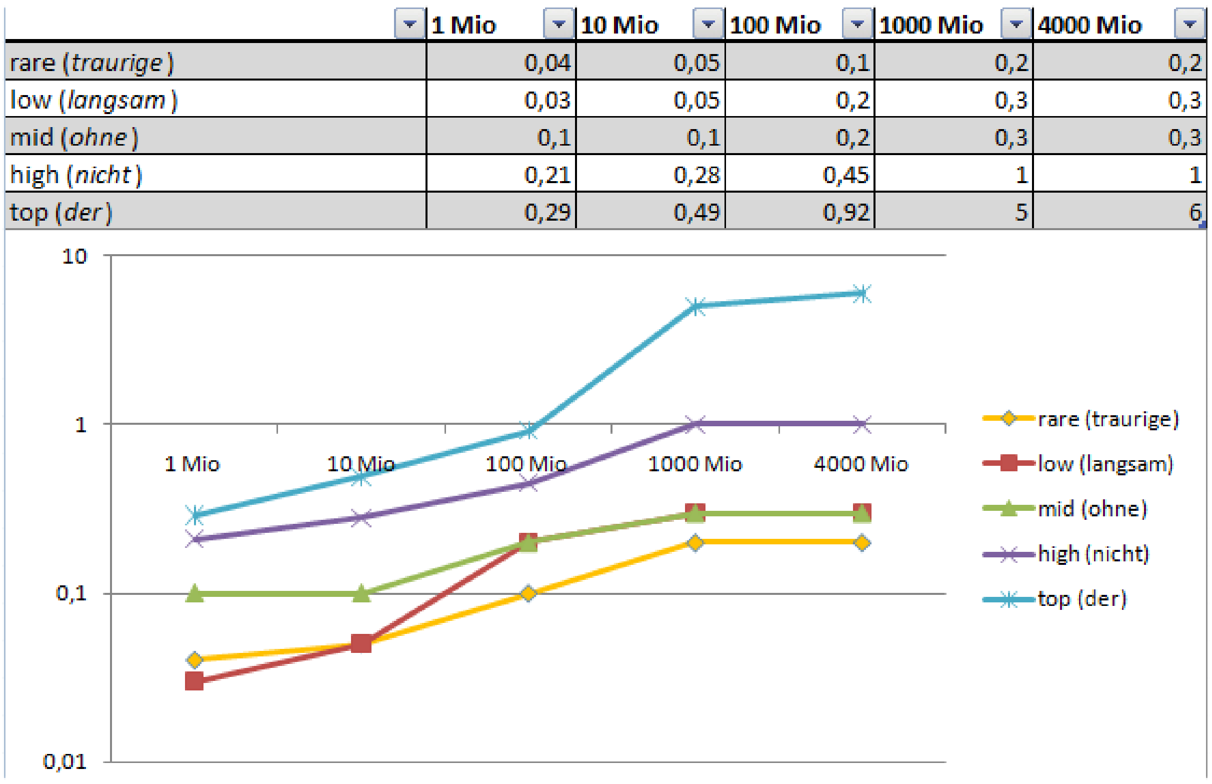
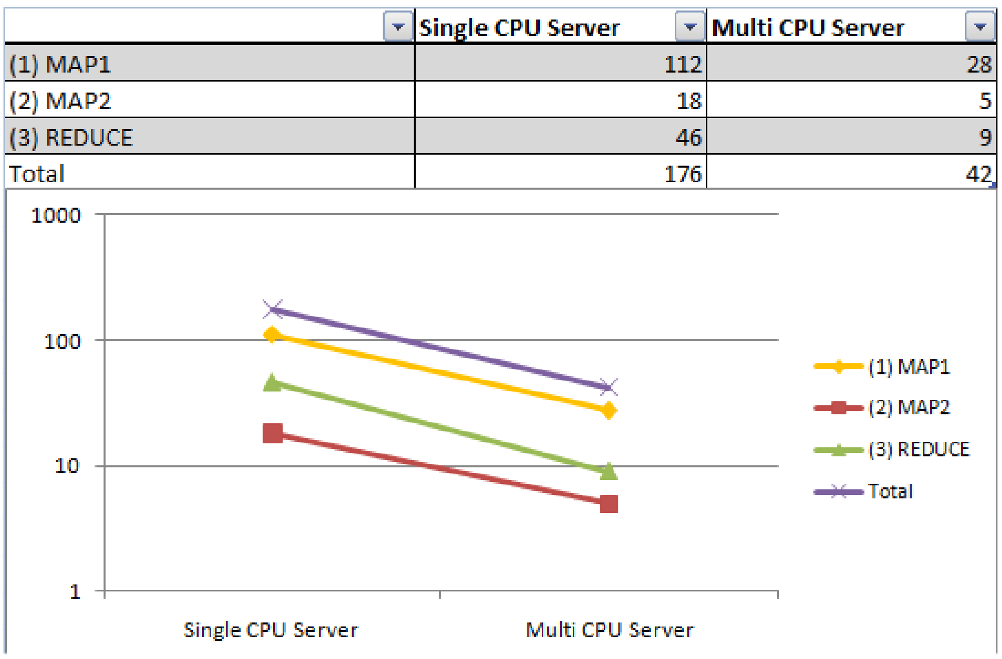


Figure 5: Response times (s) for nested SQL queries with two search keys (logarithmic scaled axis).



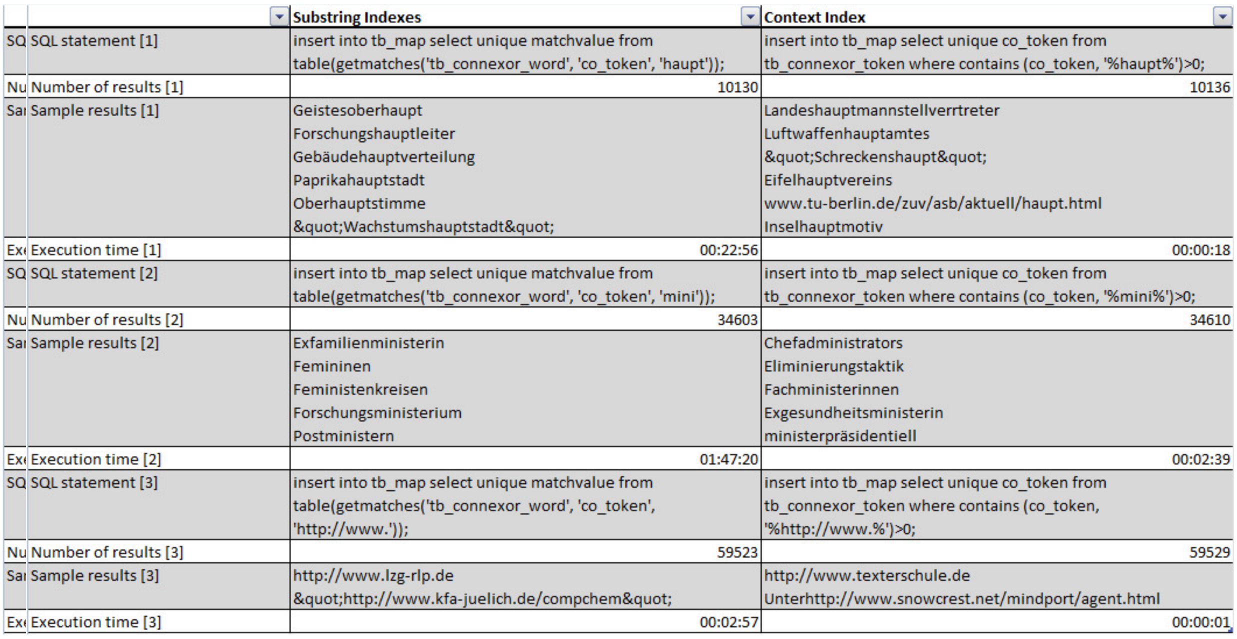
	1 Mio	10 Mio	100 Mio	1000 Mio	4000 Mio
rare (*traurige*)	0,04	0,05	0,1	0,2	0,2
low (*langsam*)	0,03	0,05	0,2	0,3	0,3
mid (*ohne*)	0,1	0,1	0,2	0,3	0,3
high (*nicht*)	0,21	0,28	0,45	1	1
top (*der*)	0,29	0,49	0,92	5	6

Figure 6: Response times (s) for nested SQL queries with one search key (logarithmic scaled axis).



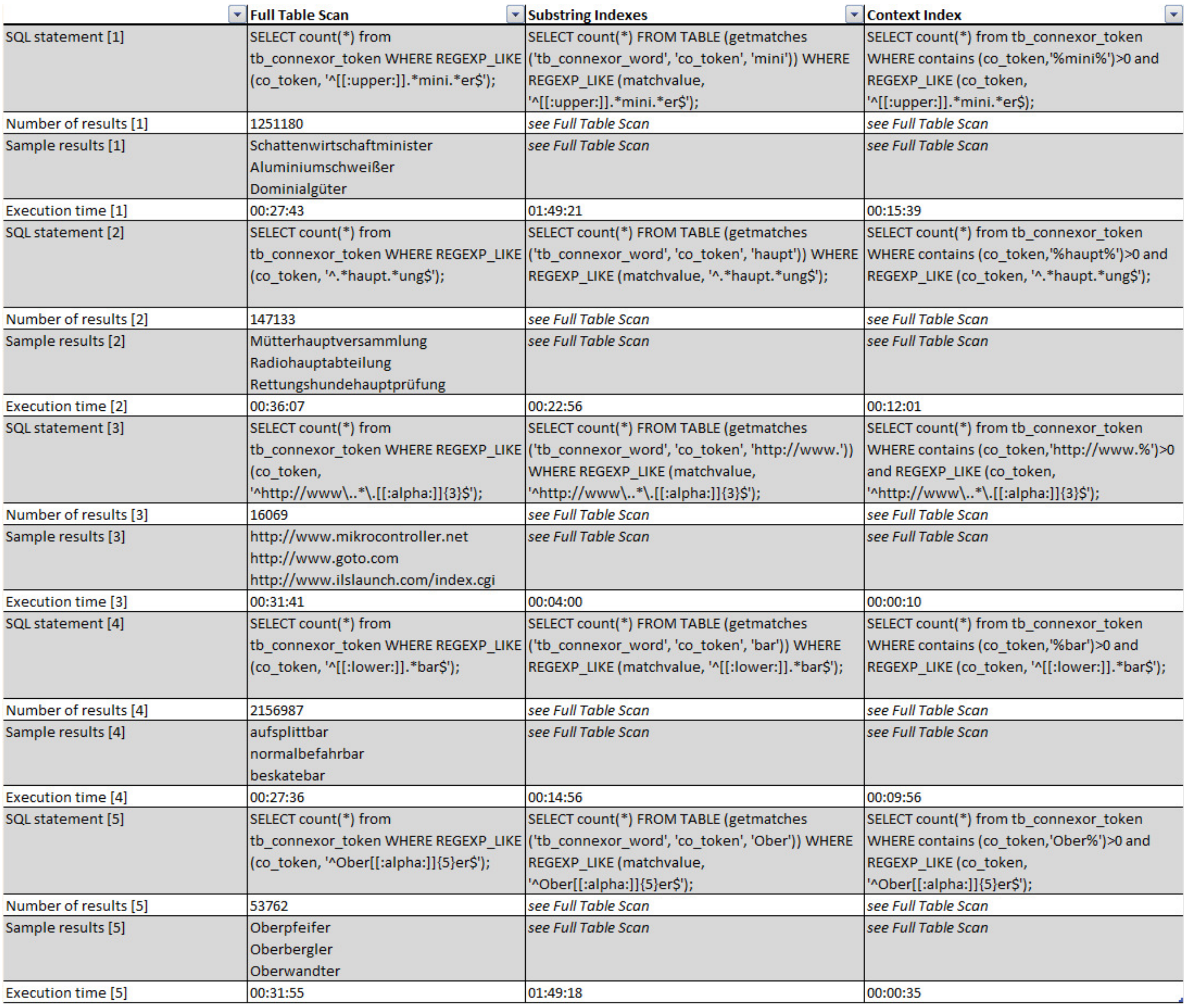
	Single CPU Server	Multi CPU Server
(1) MAP1	112	28
(2) MAP2	18	5
(3) REDUCE	46	9
Total	176	42

Figure 7: Execution times (s) for MapReduce processes on the 4-billion word corpus with three top frequent search keys (logarithmic scaled axis).



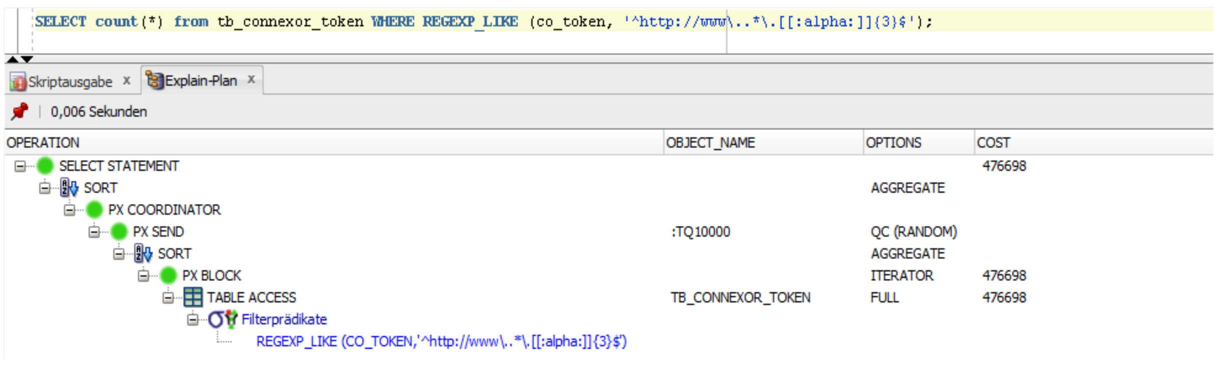
		Substring Indexes	Context Index
SQ	SQL statement [1]	insert into tb_map select unique matchvalue from table(getmatches('tb_connexor_word', 'co_token', 'haupt'));	insert into tb_map select unique co_token from tb_connexor_token where contains (co_token, '%haupt%')>0;
Nu	Number of results [1]	10130	10136
Sa	Sample results [1]	Geistesoberhaupt Forschungshauptleiter Gebäudehauptverteilung Paprikahauptstadt Oberhauptstimme &quot;Wachstumshauptstadt&quot;	Landeshauptmannstellverrtreter Luftwaffenhauptamtes &quot;Schreckenshaupt&quot; Eifelhauptvereins www.tu-berlin.de/zuv/asb/aktuell/haupt.html Inselhauptmotiv
Ex	Execution time [1]	00:22:56	00:00:18
SQ	SQL statement [2]	insert into tb_map select unique matchvalue from table(getmatches('tb_connexor_word', 'co_token', 'mini'));	insert into tb_map select unique co_token from tb_connexor_token where contains (co_token, '%mini%')>0;
Nu	Number of results [2]	34603	34610
Sa	Sample results [2]	Exfamilienministerin Femininen Feministenkreisen Forschungsministerium Postministern	Chefadministrators Eliminierungstaktik Fachministerinnen Exgesundheitsministerin ministerpräsidentiell
Ex	Execution time [2]	01:47:20	00:02:39
SQ	SQL statement [3]	insert into tb_map select unique matchvalue from table(getmatches('tb_connexor_word', 'co_token', 'http://www.'));	insert into tb_map select unique co_token from tb_connexor_token where contains (co_token, '%http://www.%')>0;
Nu	Number of results [3]	59523	59529
Sa	Sample results [3]	http://www.lzg-rlp.de &quot;http://www.kfa-juelich.de/compchem&quot;	http://www.texterschule.de Unterhttp://www.snowcrest.net/mindport/agent.html
Ex	Execution time [3]	00:02:57	00:00:01

Figure 8: Results and execution times (hh:mm:ss) for sample queries using the two different prefilter indexes.



	Full Table Scan	Substring Indexes	Context Index
SQL statement [1]	SELECT count(*) from tb_connexor_token WHERE REGEXP_LIKE (co_token, '^[[:upper:]].*mini.*er$');	SELECT count(*) FROM TABLE (getmatches ('tb_connexor_word', 'co_token', 'mini')) WHERE REGEXP_LIKE (matchvalue, '^[[:upper:]].*mini.*er$');	SELECT count(*) from tb_connexor_token WHERE contains (co_token,'%mini%')>0 and REGEXP_LIKE (co_token, '^[[:upper:]].*mini.*er$);
Number of results [1]	1251180	*see Full Table Scan*	*see Full Table Scan*
Sample results [1]	Schattenwirtschaftminister Aluminiumschweißer Dominialgüter	*see Full Table Scan*	*see Full Table Scan*
Execution time [1]	00:27:43	01:49:21	00:15:39
SQL statement [2]	SELECT count(*) from tb_connexor_token WHERE REGEXP_LIKE (co_token, '^.*haupt.*ung$');	SELECT count(*) FROM TABLE (getmatches ('tb_connexor_word', 'co_token', 'haupt')) WHERE REGEXP_LIKE (matchvalue, '^.*haupt.*ung$');	SELECT count(*) from tb_connexor_token WHERE contains (co_token,'%haupt%')>0 and REGEXP_LIKE (co_token, '^.*haupt.*ung$');
Number of results [2]	147133	*see Full Table Scan*	*see Full Table Scan*
Sample results [2]	Mütterhauptversammlung Radiohauptabteilung Rettungshundehauptprüfung	*see Full Table Scan*	*see Full Table Scan*
Execution time [2]	00:36:07	00:22:56	00:12:01
SQL statement [3]	SELECT count(*) from tb_connexor_token WHERE REGEXP_LIKE (co_token, '^http://www\..*\.[[:alpha:]]{3}$');	SELECT count(*) FROM TABLE (getmatches ('tb_connexor_word', 'co_token', 'http://www.')) WHERE REGEXP_LIKE (matchvalue, '^http://www\..*\.[[:alpha:]]{3}$');	SELECT count(*) from tb_connexor_token WHERE contains (co_token,'http://www.%')>0 and REGEXP_LIKE (co_token, '^http://www\..*\.[[:alpha:]]{3}$');
Number of results [3]	16069	*see Full Table Scan*	*see Full Table Scan*
Sample results [3]	http://www.mikrocontroller.net http://www.goto.com http://www.ilslaunch.com/index.cgi	*see Full Table Scan*	*see Full Table Scan*
Execution time [3]	00:31:41	00:04:00	00:00:10
SQL statement [4]	SELECT count(*) from tb_connexor_token WHERE REGEXP_LIKE (co_token, '^[[:lower:]].*bar$');	SELECT count(*) FROM TABLE (getmatches ('tb_connexor_word', 'co_token', 'bar')) WHERE REGEXP_LIKE (matchvalue, '^[[:lower:]].*bar$');	SELECT count(*) from tb_connexor_token WHERE contains (co_token,'%bar')>0 and REGEXP_LIKE (co_token, '^[[:lower:]].*bar$');
Number of results [4]	2156987	*see Full Table Scan*	*see Full Table Scan*
Sample results [4]	aufsplittbar normalbefahrbar beskatebar	*see Full Table Scan*	*see Full Table Scan*
Execution time [4]	00:27:36	00:14:56	00:09:56
SQL statement [5]	SELECT count(*) from tb_connexor_token WHERE REGEXP_LIKE (co_token, '^Ober[[:alpha:]]{5}er$');	SELECT count(*) FROM TABLE (getmatches ('tb_connexor_word', 'co_token', 'Ober')) WHERE REGEXP_LIKE (matchvalue, '^Ober[[:alpha:]]{5}er$');	SELECT count(*) from tb_connexor_token WHERE contains (co_token,'Ober%')>0 and REGEXP_LIKE (co_token, '^Ober[[:alpha:]]{5}er$');
Number of results [5]	53762	*see Full Table Scan*	*see Full Table Scan*
Sample results [5]	Oberpfeifer Oberbergler Oberwandter	*see Full Table Scan*	*see Full Table Scan*
Execution time [5]	00:31:55	01:49:18	00:00:35

Figure 9: Results and execution times (hh:mm:ss) for sample RegExp queries with and without prefilter indexes.



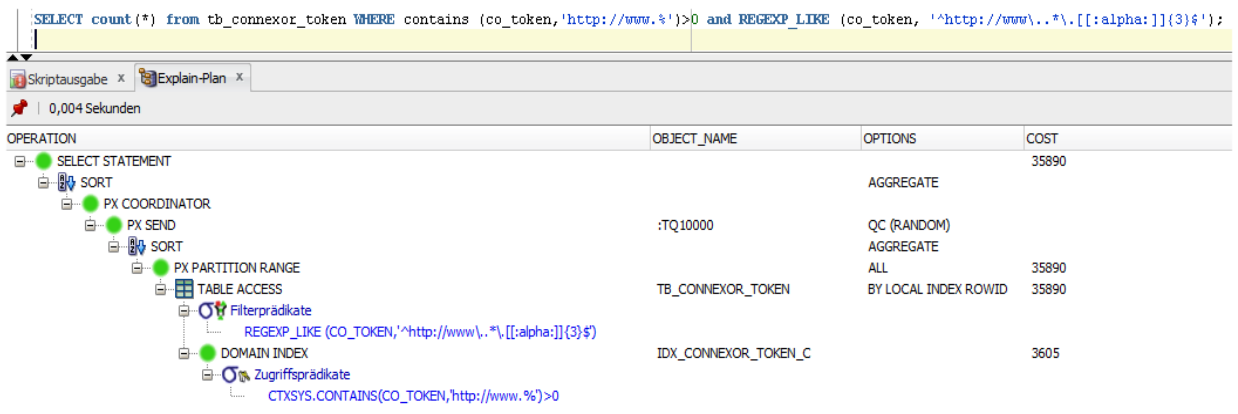
```
SELECT count(*) from tb_connexor_token WHERE REGEXP_LIKE (co_token, '^http://www\..*\.[[:alpha:]]{3}$');
```

Skriptausgabe x | Explain-Plan x

0,006 Sekunden

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			476698
SORT		AGGREGATE	
PX COORDINATOR			
PX SEND	:TQ10000	QC (RANDOM)	
SORT		AGGREGATE	
PX BLOCK		ITERATOR	476698
TABLE ACCESS	TB_CONNEXOR_TOKEN	FULL	476698
Filterprädikate			
REGEXP_LIKE (CO_TOKEN,'^http://www\..*\.[[:alpha:]]{3}$')			

Figure 10: Explain plan for sample RegExp queries without prefiltering.



```
SELECT count(*) from tb_connexor_token WHERE contains (co_token,'http://www.%')>0 and REGEXP_LIKE (co_token, '^http://www\..*\.[[:alpha:]]{3}$');
```

Skriptausgabe x | Explain-Plan x

0,004 Sekunden

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			35890
SORT		AGGREGATE	
PX COORDINATOR			
PX SEND	:TQ10000	QC (RANDOM)	
SORT		AGGREGATE	
PX PARTITION RANGE		ALL	35890
TABLE ACCESS	TB_CONNEXOR_TOKEN	BY LOCAL INDEX ROWID	35890
Filterprädikate			
REGEXP_LIKE (CO_TOKEN,'^http://www\..*\.[[:alpha:]]{3}$')			
DOMAIN INDEX	IDX_CONNEXOR_TOKEN_C		3605
Zugriffsprädikate			
CTXSYS.CONTAINS(CO_TOKEN,'http://www.%')>0			

Figure 11: Explain plan for sample RegExp queries with CONTEXT-based prefiltering.