Unlocking Language Archives Using Search

Herman Stehouwer
MPI for Psycholinguistics
herman.stehouwer@mpi.nl

Eric Auer
MPI for Psycholinguistics
eric.auer@mpi.nl

Abstract

The Language Archive manages one of the largest and most varied sets of natural language data. This data consists of video and audio enriched with annotations. It is available for more than 250 languages, many of which are endangered.

Researchers have a need to access this data conveniently and efficiently. We provide several browse and search methods to cover this need, which have been developed and expanded over the years. Metadata and content-oriented search methods can be connected for a more focused search.

This article aims to provide a complete overview of the available search mechanisms, with a focus on annotation content search, including a benchmark.

1 Introduction

Digital preservation of cultural data has been an important topic in much recent research. Large amounts of cultural heritage data is still only available in paper form. Digitization and digital preservation is relatively affordable and provides a number of significant benefits. Digital data does not degrade in quality with use. In addition, dissemination, access, and analysis techniques are easier with digital data. Furthermore, digital data enables researchers to answer questions which were unfeasible to answer before. (Ordelman et al., 2009; Reynaert, 2010)

Some of these digitization and accessibility projects have focused specifically on access, i.e., by improving search methods for the domain. Only a powerful search tool allows researchers to quickly find their "needles in the haystack". (Auer et al., 2010; Kemps-Snijders et al., 2009; Kemps-Snijders et al., 2010; Ringersma et al., 2010; Skiba, 2009; Wittenburg and Trilsbeek, 2010; Wittenburg et al., 2010; Johnson, 2002)

In terms of preservation and access to cultural heritage data the MPI occupies a unique position. The uniqueness is captured by four aspects of the situation, (1) the cultural heritage data concerns mainly currently spoken linguistic data, (2) often the linguistic data is accompanied by video to capture non-verbal communication and cultural background, (3) the archive now hosts data from many linguistic preservation projects, linguistic studies and psycholinguistic experiments, and (4) there are a variety of methods offered to browse, search, and leverage the large archive of data.

This article is structured as follows. In Section 2 we provide an overview of the archive, in the form of a brief history and an overview of available data. In Section 3 we describe the current access methods, including our powerful combined metadata and content search. We will present some performance statistics for the content search in Section 4. We end this article in Section 5 with a summary.

2 The Archive

The archive stores a large variety of material in more than 250 different languages. It contains circa $160,000$ annotation files for more than $200,000$ audio or video recordings. The recordings include more than $4,300$ hours of SD quality[1] video and more than $3,500$ hours of CIF quality[2] video. The archived content is supported by almost $200,000$ metadata files and $50,000$ auxiliary information files.

[1] SD means Standard Definition (resolution). Circa $14,500$ videos, 78% are 720×576 pixels (resolutions from $640 \ldots 768 \times 480 \ldots 576$)

[2] CIF means Common Intermediate Format, which implies a specific (lower) resolution range. Circa $40,400$ videos, 78% are 352×288 pixels (resolutions from $320 \ldots 384 \times 240 \ldots 384$, plus $2,350$ double width videos merged from 2 CIF videos each)

Proceedings of Language Technologies for Digital Humanities and Cultural Heritage Workshop, pages 19–26,
Hissar, Bulgaria, 16 September 2011.

The material in the archive occupies more than 40 terabyte of storage capacity, most of which for the media recordings. There are 22 gigabytes of annotation files, 2.5 gigabytes of metadata and 7 gigabytes of auxiliary files. In addition, there is currently more than 55 terabyte of "pre-archive" data in the pipeline on the way to the archive. Based our experience, the creation of one terabyte of archive-able data costs around 1.5 million €.

(Wittenburg et al., 2010) gives a recent description of the state of The Language Archive (TLA). They deal with three aspects pertaining to the archive: (1) replication of archived material, bitwise copies of the original material – each file is stored in six copies in two countries, (2) encoding and metadata standards, and how they apply to the archive itself, and (3) controlled access to archived materials. The article provides a good overview of the TLA resources and software used for managing the archive.

Access Methods

The archive itself is accessible in a number of ways, most of which can be reached from the www.clarin.eu Virtual Language Observatory (VLO) (Uytvanck et al., 2010). The classic access method is the IMDI[3] browser, which displays a tree view on the Directed Acyclic Graph (DAG) type archive structure graph.

When using the catalogue on the CLARIN VLO portal, the TLA archive and several external archives can be visited in one browser session. To enable central access, CLARIN exchanges and harvests metadata with other archives using the OAI protocol. We remark that the different archives are also accessible separately. For instance, the TLA data is accessible on the web at www.lat-mpi.eu/tools/browser. A faceted search (based on Apache SOLR) of the combined archives is available in the VLO language resource inventory described below. The researcher can also browse the virtual language world in the VLO. The virtual language world presents the available corpora on a world map. Faceted search and geographical browsing always present the combined archives, not just TLA data. The TLA archive is also accessible via a number of other search methods, which we outline below.

Quality Control

The whole archive is permanently under active quality control. Files can only be uploaded by authorized researchers using the web-based LAMUS[4] archive upload and editing tool which also applies format checks. Those checks ensure that only file formats for which free viewers (and preferably editors) are widely available are stored. Archive managers define format rules and make statistics about archived data, e.g. about video resolutions or audio sampling rates. They also run regular consistence checks on the archive, link structure and file formats.

For annotation file formats supported by the search methods, files are parsed to verify their syntactic validity. Parse logs are reviewed to find problematic files and adapt parsers for common syntax variants, for example for CHAT, ELAN and Toolbox[5]. The customized parser for example has heuristical parsing of CHAT participant lists, which in turn can be used to search.

3 Searching

We aim to help researchers answer their scientific questions. Often these questions may be answered by providing the right data. To help locate the right data we provide a variety of browse and search methods. In this section we aim to give a brief overview of each of the available methods. Any part or multiple parts of the DAG tree in the IMDI browser can be selected to perform any type of search on.

3.1 Metadata Search

Here we briefly describe the metadata search. The metadata search is available from within the IMDI browser. The metadata search contains two different search methods: (1) a keyword search, and (2) an advanced metadata search.

The first performs a quick search for a set of keywords in all available metadata fields. The second allows the researcher to define a set of constraints. These constraints are then used to select all matching records in the part of the archive that is searched on.

[3] Isle MetaData Initiative, an XML metadata standard.

[4] Language Archive Management and Upload System – www.lat-mpi.eu/tools/lamus

[5] CHILDES CHAT: Child Language Data Exchange System, Codes for the Human Analysis of Transcripts childes.psy.cmu.edu/clan/. ELAN EAF: EUDICO Linguistic Annotator www.lat-mpi.eu/tools/elan/. Toolbox, Shoebox: www.sil.org/computing/toolbox/.

We provide a telling example. To search for audio recordings of young speakers, we can use the following two constraints: (1) actor age is smaller than 10, and (2) the format of the resource is an audio file. The second constraint is entered with a user friendly choice list. Optionally, the researcher can see which choices would match how often. Using a fast Apache Digester index, results are presented without noticeable delay.

Once a researcher has performed a metadata search they can choose to use the results in three ways: (1) the results can be viewed and printed, (2) the results can be exported as links in an IMDI file for later use, and (3) the results can be used directly to specify the domain of a content search (described below).

3.2 Trova Annotation Content Search

Here we briefly describe Trova, the annotation content search. A Trova search always starts from a selection of elements in the archive, which are used as the search domain. Typically the search domain consists of a corpus, or of the domain selected using the metadata search. While all metadata is freely accessible, all access to annotation content including search is controlled by the access rules[6] for the individual corpora.

Trova supports three different search methods: (1) the simple search, (2) the single layer search, and (3) the multiple layer search. We describe them in order of complexity.

In all three search modes, the researcher can select which of the searchable file types should be considered: ELAN EAF, CHILDES CHAT, Shoebox, Toolbox, text, HTML, XML, PDF, SubRip, Praat TextGrid and CSV[7].

Usage Considerations

The Trova application is the main way to search the online archive. However, after excluding queries made for demonstration, teaching and testing purposes, it turned out that Trova was not used heavily in the past:

In the period from April 2008 to July 2010, there were more than 2000 user queries, about 80 per month, of which 75% unique. Most searches were in Dutch corpora such as CGN. Simple or single layer search were most common.

In the first half of the analyzed period, only 11 queries per month used structured multi layer search. Most structured search queries had a complexity of up to four keywords or constraints and were not in Dutch corpora. Half of the complex queries used a constraint that keywords in two tiers had to co-occur (same timing).

Two possible reasons for the infrequent use of Trova in the past are the steep learning curve of structured search and slow speed. To address this, we improve documentation and teaching. Also, Trova was slow compared to typical web search engines. Incremental processing already helped by showing the first results while the query was still running. However, overall processing time was still high and complete result lists are of more interest in linguistic context than in web searches.

Optimizations

To optimize search performance, searching is performed in three steps: First, when a researcher enters the search page, the properties of all tiers (a tier is a layer of annotation) in the selected search domain are processed. Second, as soon as a query is made, fingerprinting is used to limit searching to a small set of candidate tiers to improve query speed. Each tier is indexed with four different character n-gram fingerprints. The current fingerprints group all possible combinations of 1 to 4 characters into a limited number of slots. A tier can only contain a match for a given keyword when it fills at least all slots used by that keyword. For regular expression search, plain text is extracted from the query to still use partial fingerprinting. Third, the candidate tiers are processed in small groups, displaying hits as they come in.

While a query is still running, the researcher can already browse the first results. However, the hits will only be displayed in their final ordering when they all have been found. At that point, hits can also be viewed sorted by their most frequent concordances. Hits can be saved in CSV files using a user-selectable order and choice of columns. From each hit, the researcher can navigate to Annex (Berck and Russel, 2006), a browser based presentation of the parsed annotation file along with corresponding media files.

[6] The TLA AMS access management system (www.lat-mpi.eu/tools/ams) allows to define individual and group access rules on the level of corpora, sessions, filetypes and files.

[7] Our database contains circa $123,000,000$ annotations in $750,000$ tiers from $110,000$ parsed files. The most common annotation file types are CHAT ($48,600$ files), EAF ($28,100$ files) and plain text ($26,400$ files).

Simple Search

The simple search allows searching for keywords in the selected search domain. The search performed using these keywords performs a case-insensitive substring matching.

Single Layer Search

Single layer search gives the researcher more control over the search than when using the simple search. The researcher can select whether matching should use exact matching, substring matching, or regular expression matching. Furthermore the researcher can choose whether he wants the matching to be case sensitive. It is also possible to perform searches over n-grams of annotations. Both n-grams inside and across annotations can be searched. The n-gram search modes support single position wildcards (#) and per word negation, e.g., *the # NOT(green) house*. In the example, the phrase *I went to the big red house yesterday.* would match.

We remark that exact match means that one annotation has to match the keyword exactly. The researcher has to be aware that some annotation tiers annotate whole utterances as one annotation while others annotate word by word. In some cases, searching with a regular expression can be more appropriate.

A further option in single layer search is restricting the search to a subset of the available tiers. Annotation tiers can be selected by several properties, namely: name, type, participant, and annotator. The researcher can see how many tiers match which value and can sort the choice list by that. This allows to quickly see that a corpus contains more *type pho* tiers than *type Phonetic* tiers[8].

Multiple Layer Search

Multiple layer search is the most advanced search interface available on the archive. In multiple layer search, a grid of search terms can be entered, with constraints between them. Constraints on the X axis can be used to require a certain (or minimum or maximum) time or number of annotations between keyword hits. Constraints on the Y axis give the researcher fine grained control over whether and how keyword hits in different tiers have to overlap. In the grid, the X axis corresponds to the time axis, while the Y axis corresponds to different tiers within one file. Similar to the single layer search, the researcher can activate constraints about the properties of tiers, but now separately for each grid row. An example query could be *'pink' before 'elephant' in a text type tier, 'elephant' overlapping 'big' time-wise in a gesture type tier*. In Figure 1 we show an example of the multiple layer search, showing this example.

3.3 CQL Search Service

In the context of the European search infrastructure we make available a machine-accessible web-service for content search. This web-service provides a search-service on parts of the archive (as access rights permit) and is integrated in the European search infrastructure. The European search infrastructure provides a central location for researchers to perform searches in many different language resources. The web-service is based on SRU/CQL(Morgan, 2004; S.H. McCallum, 2006), where SRU is the communication protocol and CQL the search query language.

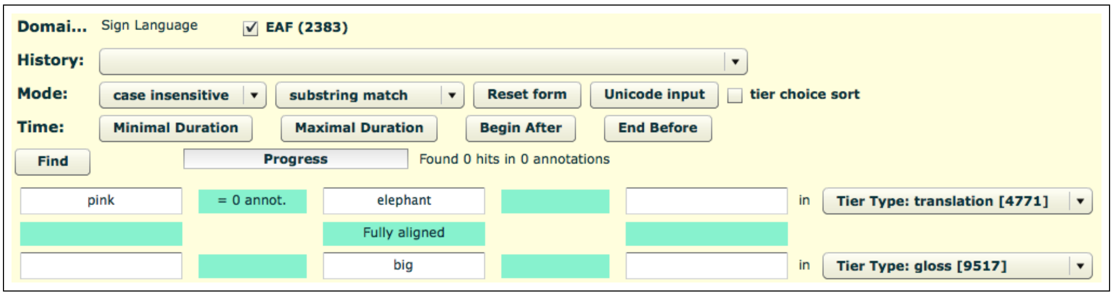
CQL search provides functionality based on the Trova single layer search through a REST[9] interface. More complex processing will be added in the future, as CQL allows to express fairly complex queries. These queries differ from the 2d grid paradigm of Trova. As stateless REST means processing whole queries in one single HTTP access, CQL search can cache intermediate results for later re-use. So when the same researcher makes multiple queries to the same corpus, hits will be returned faster. REST queries can optionally return as soon as some results have been found or wait until all results are ready.

3.4 Virtual Language Observatory

A separate way in which to browse the metadata is available in the Virtual Language Observatory (VLO). The VLO makes available a faceted browser on the metadata from several language sources, including our archive. To use the VLO, enter the *language resource inventory* on the www.clarin.eu/vlo virtual language observatory page. In faceted browsing, different IMDI and CMDI metadata fields can be used to zoom in on corpora. Supported facets include origin (e.g., MPI, Open Language Archives Community), con-

[8] Unfortunately, there are no widely used controlled vocabularies to classify tiers. We plan to use ISOcat / RELcat concept registries to allow a more semantic view on tier properties. This would allow selections such as *tier types with parent category DC-2641: phonetics*.

[9] Representational State Transfer, e.g. via HTTP.


Domai... Sign Language ☑ EAF (2383)

History:

Mode: case insensitive ▾ substring match ▾ □ tier choice sort

Time:

Found 0 hits in 0 annotations

pink = 0 annot. elephant in Tier Type: translation [4771] ▾

Fully aligned

big in Tier Type: gloss [9517] ▾

Figure 1: Partial screenshot of multiple layer search. Showing the *pink elephant* search example.

tinent, country, language, resource type (e.g., audio), subject, genre, and organization.

All facet lists are shown with counts of occurrences which are dynamically updated as the researcher makes selections. For example, after selecting Dutch ($18,000$ resources), the facet origin is updated, showing that most Dutch resources are from the CGN corpus. The researcher can then proceed to specify more facets, for example select the Dutch Bilingualism Database (DBD) corpus as origin, select a genre, and so on. Facets can be specified in any order and page updates are almost instantaneous, using a SOLR database.

At any moment, a keyword search on metadata descriptions can be used to narrow down the results. From the result list, metadata sets can be displayed as tables and the researcher can jump directly to resources such as audio recordings.

3.5 ELAN Structured Multiple File Search

A modified version of Trova is one of the search functions of ELAN annotation editor. This enables the researcher to search in (a group of) annotation files while they are still being worked on. Different from Trova, this search parses files on the fly. ELAN can import and export a number of other file formats and the search could parse some of them directly. In addition, ELAN supports list-of-constraints style search similar to metadata search. Not using a full corpus database means reduced speed compared to Trova, but all annotation updates are available in searches immediately.

4 Annotation Search Benchmark

4.1 Test Corpus and Hardware

The various metadata search methods provide results almost instantaneously. However, with almost $120,000,000$ individual annotations in more than $100,000$ annotation files in the archive, content search is a more demanding task. Thanks to various optimizations described above, such as fingerprinting of tiers and queries, response times are still reasonably fast.

To provide some benchmarks, we ran a number of searches on a large subset of the archive consisting of several sizable corpora: All of DoBeS[10], the Dutch Spoken Corpus CGN, the MoLL L2 acquisition corpus[11] as well as local mirrors of Talkbank and the CHILDES sub-corpora Biling, Japanese, Cantonese, Turkish, Spanish, French and German. The size of this search space is almost $55,000,000$ annotations in $354,000$ tiers in $43,000$ files. The initial analysis of the search domain can take 20 seconds or more but then multiple queries can be done with low further overhead. Finding **all** occurrences of *elephant*, *needle* or *haystack* in more than 50 million annotations can then be done within 7 seconds and initial results are shown much sooner. Among other things, speed depends on narrowing down the search space by fingerprinting and on I/O caching at the Linux and PostgreSQL level. Searching in smaller corpora is much faster. For example searching only in the seven mentioned CHILDES sub-corpora, worth about $4,000,000$ annotations, has a set-up time of less than 10 seconds and after that, searches take at most a few seconds.

All benchmarks were done on an older test server with two dual core 1.8 GHz AMD Opteron 265 CPUs, 16 GB of DDR400 RAM (8 modules) and a small SCSI 160 RAID with 120 MB/s read bandwidth. A modern desktop PC can have twice the speed, cores, RAM and disk bandwidth with one CPU and a consumer SSD. Trova used up to three database threads on our test server, PostgreSQL uses one core per thread. PostgreSQL is configured to use 3.5 GB of shared buffers, 1 GB of work memory per task and 6 GB of cache size.

[10] Dokumentation Bedrohter Sprachen

[11] Project “Modalität in Lernervarietäten im Längsschnitt”

4.2 Task Design

We searched for ten keywords each from a set of eight languages in our 55M annotation test domain. For German, English and Dutch, we used entries 991 to 1000 of the “top1000” lists of wortschatz.uni-leipzig.de; For French, Spanish and Turkish, we used entries 991 to 1000 of the word frequency lists on en.Wiktionary.org; For Russian, the masterrussian.com list of most common words was used. No frequency list was available for Japanese, so we used a selection of words from “1000 Japanese basic words” from Wiktionary. The Japanese selection contains nouns which are translations of words present in the selections of the other languages.

For this set of terms[12], we investigated three search paradigms in sequence: (1) keyword search with substring matching (i.e., the keyword can match any part of the annotation), (2) regular expression search, either for “word between word boundaries” (German, English, Dutch, French and Spanish), or for “word starts with…” (Turkish, Russian and Japanese)[13], and (3) keyword search with exact matching (i.e., the keyword must match the entire annotation).

All queries were done via the CQL REST interface three times in a row, requesting to wait until **all** results have arrived. We observe that repetitions of queries differ in average speed by less than 10%. In addition, we first searched for a bogus word (**start**), taking up to 20 seconds while CQL search caches domain properties before the timed queries start.

4.3 Results

Substring Search

In Table 1 we show the results for the substring matching. We observe that the query speed varies significantly with language: Our test corpus contains a large number of Dutch annotations, so this task takes the most time and finds the most hits. Searching for English, French, German and Spanish already is twice as fast, as is Turkish. Turkish words tend to have fingerprints similar to those of words in other languages. Searching only in the Turkish sub-corpus would be a lot faster.

Block	AM	AM	MD	Min	Max
substr	hits	duration (in seconds)			
Dutch	9453	13.16	13.23	4.2	27.5
English	3532	8.26	6.44	4.2	20.0
French	3603	7.25	6.19	3.9	17.4
German	1634	6.57	4.75	2.7	23.0
Japanese	689	0.89	0.55	0.3	2.0
Russian	50	0.61	0.60	0.5	0.8
Spanish	1336	5.98	5.10	4.1	9.4
Turkish	113	6.80	6.75	2.1	23.1

Table 1: Benchmark results for substring queries. 30 queries per language, 60 for Japanese. AM = Arithmetic Mean, MD = Median.

Block	AM	AM	MD	Min	Max
regexp	hits	duration (in seconds)			
Dutch	2834	11.55	10.86	4.7	25.8
English	2512	9.18	7.71	5.1	26.2
French	2134	7.68	7.60	3.7	21.6
German	371	5.85	4.77	2.8	11.6
Japanese	414	0.85	0.51	0.3	1.9
Russian	15	0.64	0.63	0.5	0.9
Spanish	882	5.55	4.81	3.8	8.1
Turkish	132	9.56	7.84	3.4	32.2

Table 2: Benchmark results for regexp queries. 30 queries per language, 60 for Japanese. AM = Arithmetic Mean, MD = Median.

But why do we get all results within less than one second for Japanese and Russian, without explicitly searching only in relevant sub-corpora? This is again due to fingerprinting: Text in other languages will most likely not contain any cyrillic, kanji or hiragana characters at all. So even by looking only at unigrams, Trova and CQL search can quickly discard most tiers in other languages when searching for Japanese or Russian words.

Regular Expression Search

The second block of queries uses regular expression search, results are shown in Table 2. As expected, we observe fewer hits than with a plain substring search. This also explains small speed

[12] We replaced a number of word forms from the original lists to be more suitable for raw string search as follows: *pahnut’* (infinitive) to *pahnet*, *zavod’* (-’ only in one form) to *zavod*, *querías* to *quería*, *hareket etmek* (inf.) to *hareket ettiler* and *istenmek* (inf.) to *istenem*. Of course we used cyrillic strings in the actual queries. The latin transliterations are only used for easier reading.

[13] Word boundaries ($\backslash$*bkeyword*$\backslash$*b*) are ASCII-oriented, not covering accented characters, but do work with punctuation marks. The regular expression $(\backslash s|\backslash A)$*keyw* works in Unicode space, anchoring *keyw* to the start of the annotation or a space. However, e.g. opening parentheses or quotation marks before *keyw* will not match.

Block	AM	AM	MD	Min	Max
exact	hits	duration (in seconds)			
Dutch	1756	10.25	9.18	3.4	26.8
English	64	7.02	5.44	3.6	18.7
French	45	5.91	5.12	3.5	12.7
German	3	5.66	3.88	2.4	22.9
Japanese	0	0.74	0.40	0.2	2.8
Russian	6	0.51	0.50	0.4	0.7
Spanish	26	4.87	4.26	3.5	7.0
Turkish	1	5.23	5.58	2.1	7.9

Table 3: Benchmark results for exact queries. 30 queries per language, 60 for Japanese. AM = Arithmetic Mean, MD = Median.

Block	AM	AM	MD	Min	Max
all	hits	duration (in seconds)			
Dutch	4681	11.65	10.29	3.4	27.5
English	2036	8.15	6.61	3.6	26.2
French	1927	6.95	5.58	3.5	21.6
German	669	6.03	4.51	2.4	23.0
Japanese	368	0.83	0.52	0.2	2.8
Russian	24	0.58	0.59	0.4	0.9
Spanish	748	5.47	4.92	3.5	9.4
Turkish	82	7.19	6.43	2.1	32.2

Table 4: Benchmark results for all queries. 30 queries per language, 60 for Japanese. AM = Arithmetic Mean, MD = Median.

gains for Dutch and Spanish. For the other languages, in particular Turkish, a small speed loss can be seen. Here, two forces act on the processing load: Searching for (shorter) prefixes means that fewer tiers can be discarded based on n-gram fingerprints. More have to be considered, yet those contain fewer hits because specifying a regular expression is more restrictive than a substring. In addition, regular expressions take more CPU time to process. Note that the fingerprinting only considers the plain parts: For example, a search for $(\backslash s|\backslash A)$*yumurt* will consider all tiers which satisfy the n-grams of *yumurt*[14].

Exact String Search

Our third round of queries only considers exact matches. We show this round in Table 3. While there is some speed gain compared to substring matches, related to having fewer hits, it is much smaller than expected. Some time is saved because string inequality can often be detected without having to scan the whole string. However, the set of candidate tiers chosen by fingerprinting is as big as for substring search.

Adding specific indexes can improve exact match speed, but will only have an effect on this match mode. For example, such an index could bin whole-string hashes in slots. Another possibility would be an index of only the sets of string lengths occurring in each tier.

Substring searches are most used, especially because not all corpora have annotations at word granularity. Many corpora annotate larger units, such as phrases, sentences or utterances, but at higher quality, e.g., stating recording timestamps for them. Searching for annotations which are exactly *elephant* will not work in a corpus treating *I saw an elephant.* as one atomic annotation.

Overall Speed

Finally, Table 4 gives a summary of all queries in our benchmark task: The average and in particular median query durations in our 55M annotation test corpus are considerably below 10 seconds for most languages. For languages which can be readily identified from their writing system, waiting times below one second can be expected.

While it is not visible in this table, our experience shows that long waiting times relate to novel queries for hard to filter words. The slowest query is the regular expression search for words starting with *isten*. Searching for *isten*-after-space would be faster (37% fewer candidate tiers) but would not find annotation-initial occurrences of *isten*. Repeating the query later reduces waiting time from 32 to 28 seconds, showing the effects of disk caching.

5 Summary

In this article we have described the TLA language archive and possibilities to search in it. The archive contains a large and diverse collection of language data occupying over 40 terabytes of storage for more than 250 languages.

[14] The "stemming" of *yumurta* to *yumurt* is an artificial example and not meant to be linguistically correct. Using fingerprint tables up to n-gram size 4, from *y, u. . . , yu, um. . .* to *murt* have to be present in a tier to make it a candidate. To balance disk space usage against speed, not all possible combinations of 1 to 4 Unicode characters are fingerprinted separately. Instead, n-grams are hashed into bins – for example, all possible 4-grams share $2,000$ classes.

We presented a variety of browse and search methods available for our archive, developed over several years. We described the speed of our annotation content search on a small server, using a large test corpus. We have listed results from benchmarks, and analyzed them.

Furthermore, we have discussed several current and future optimizations that can improve search and browse speed. Of course, our implementation is also guided by the most common use cases.

References

[Auer et al., 2010] Auer, E., Wittenburg, P., Sloetjes, H., Schreer, O., Masneri, S., Schneider, D., and Tschöpel, S. (2010). Automatic annotation of media field recordings. In Sporleder, C. and Zervanou, K., editors, *Proceedings of the ECAI 2010 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2010)*, pages 31–34, Lisbon. University de Lisbon.

[Berck and Russel, 2006] Berck, P. and Russel, A. (2006). Annex – a web-based framework for exploiting annotated media resources. In *LREC*.

[Johnson, 2002] Johnson, H. (2002). The archive of the indigenous languages of latin america: Goals and visions. In *Proceedings of the Language Resources and Engineering Conference*, Las Palmas, Spain.

[Kemps-Snijders et al., 2010] Kemps-Snijders, M., Koller, T., Sloetjes, H., and Verweij, H. (2010). Lat bridge: Bridging tools for annotation and exploration of rich linguistic data. In Calzolari, N., Maegaard, B., Mariani, J., Odijk, J., Choukri, K., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 2648–2651. European Language Resources Association (ELRA).

[Kemps-Snijders et al., 2009] Kemps-Snijders, M., Windhouwer, M., and Wittenburg, P. (2009). Isocat: Remodeling metadata for language resources. In *International Journal of Metadata, Semantics and Ontologies (IJMSO)*, volume 4 (4), pages 261–276.

[Morgan, 2004] Morgan, E. (2004). An introduction to the Search/Retrieve URL Service (SRU). *Ariadne*.

[Ordelman et al., 2009] Ordelman, R. J. F., Heeren, W. F. L., de Jong, F. M. G., Huijbregts, M. A. H., and Hiemstra, D. (2009). Towards affordable disclosure of spoken heritage archives. *Journal of Digital Information*, 10(6):17.

[Reynaert, 2010] Reynaert, M. (2010). Character confusion versus focus word-based correction of spelling and ocr variants in corpora. *International Journal on Document Analysis and Recognition*, pages 1–15. 10.1007/s10032-010-0133-5.

[Ringersma et al., 2010] Ringersma, J., Zinn, C., and Koenig, A. (2010). Eureka! user friendly access to the mpi linguistic data archive. In *SDV - Sprache und Datenverarbeitung/International Journal for Language Data Processing*.

[S.H. McCallum, 2006] S.H. McCallum (2006). A look at new information retrieval protocols: Sru, opensearch/a9, cql, and xquery. In *WORLD LIBRARY AND INFORMATION CONGRESS: 72ND IFLA GENERAL CONFERENCE AND COUNCIL*, Seoul, Korea.

[Skiba, 2009] Skiba, R. (2009). Korpora in der Zweitspracherwerbsforschung: Internetzugang zu Daten des ungesteuerten Zweitspracherwerbs. In Ahrenholz, B., Bredel, U., Klein, W., Rost-Roth, M., and Skiba, R., editors, *Empirische Forschung und Theoriebildung: Beiträge aus Soziolinguistik, Gesprochene-Sprache- und Zweitspracherwerbsforschung: Festschrift für Norbert Dittmar*, pages 21–30.

[Uytvanck et al., 2010] Uytvanck, D. V., Zinn, C., Broeder, D., Wittenburg, P., and Gardelleni, M. (2010). Virtual language observatory: The portal to the language resources and technology universe. In Calzolari, N., Maegaard, B., Mariani, J., Odijk, J., Choukri, K., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 900–903. European Language Resources Association (ELRA).

[Wittenburg and Trilsbeek, 2010] Wittenburg, P. and Trilsbeek, P. (2010). Digital archiving – a necessity in documentary linguistics. In Senft, G., editor, *Endangered Austronesian and Australian Aboriginal languages: Essays on language documentation, archiving and revitalization*, pages 111–136. Canberra: Pacific Linguistics.

[Wittenburg et al., 2010] Wittenburg, P., Trilsbeek, P., and Lenkiewicz, P. (2010). Large multimedia archive for world languages. In *Proceedings of the 2010 ACM Workshop on Searching Spontaneous Conversational Speech, Co-located with ACM Multimedia 2010*, pages 53–56. Association for Computing Machinery, Inc. (ACM).