A Ten-Year Summary of a SOA-based Micro-services Infrastructure for Linguistic Services

Marco Büchler
mbuechler@etrap.eu
University of Goettingen, Germany

Greta Franzini
gfranzini@etrap.eu
University of Goettingen, Germany

Emily Franzini
efranzini@etrap.eu
University of Goettingen, Germany

Thomas Eckart
teckart@informatik.uni-leipzig.de
Universität Leipzig, Germany

Introduction

In the mid 1990s, the Natural Language Processing Group at the University of Leipzig began work on the Wortschatz project which aims to provide corpora in hundreds of languages and in different size-normalisations, be that 100K, 300K or 1M sentences. As the resources grew in size, so did the number of requests for the data. In the early stages of the project a specific dump was created, parts of which even came with a small user-interface. The database dump was shared with interested researchers and partners in the business sector.

After some time, however, the personnel costs of this kind of collaboration became unsustainable. For this reason, a new plan was put into motion in 2004, consisting of the development of a SOAP-based API - the *Leipzig Linguistic Services* (LLS) - that enabled any interested person to access the data of the *Wortschatz* databases in any provided language (Quasthoff et al. 2006, Eckart et al. 2012). Overall 20 services were provided, delivering specific information such as baseform, category classifications, and thesaurus data. The aim of the LLS was to establish a *Service Oriented Architecture* (SOA) for linguistic resources based on small and atomic micro-services that could be combined by users for particular needs. Users were then not only able to browse through the *Wortschatz* website, but also to integrate those services with their own existing digital ecosystems.

In 2005 these services were made publicly available and by September 2006 all requests were systematically logged. In July 2014 the number of logged requests reached nearly one billion. While at the beginning the use was limited to academia, over time the services were increasingly used by the private and business sectors as well.

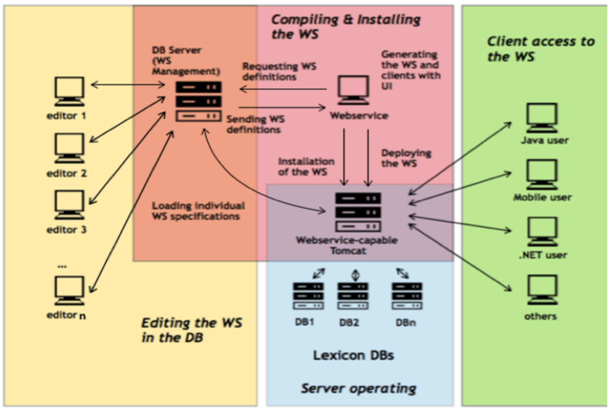


Figure 2. Four workflow modes with separation of concern: editing (yellow); managing, compiling and deploying (red); hosting and operating (blue); using the LLS infrastructure (green).

The Leipzig Linguistic Services

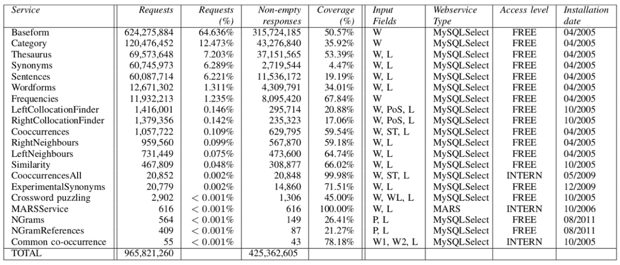
The intention of the overall LLS architecture was to be as simple and generic as possible. A generic architecture can be reused in different scenarios but tends to have too many parameters and options, while a simple architecture claims usability and guarantees a faster learning curve. In the following, we briefly describe the architecture of the LLS.

In order to create the server-side Java code for a specific webservice, a data-set needed to be added to the webservice management (yellow zone in figure 1). The necessary edits contain, besides others, information on the name and type of the webservice (see also table 1) or parameters. *Apache Ant* was used as the central tool for generating the back-end services and deploying them in a *Tomcat* server (see red zone in figure 1). The blue zone illustrates the operations of the *Wortschatz* databases. Using the generic description of the webservice in the WSDL-files a number of wrappers of generated source code were created and made publicly available by LLS users such as for C# as part of .NET, Perl, Python, Delphi, PHP, Ruby and JavaScript (see green zone in figure 1).

Independently from the underlying programming languages, over the past ten years we have observed different uses in research, business and in the private sector. In research, the LLS were used in the areas of text profiles and author classification (Borchardt 2005). The services were also used as data resources for sentiment analysis or for query expansion. Users from the business field were mainly interested in using *Baseform* or *Synonym* services for improving internal search indexes. The LLS data was also used for information retrieval tasks in portals for weighting words in a word cloud or to display enriching information. Private users accessed the LLS to complete crossword puzzles. A dedicated service was installed upon request just for this purpose (see also table 1), since it was possible to query a pattern of an incomplete word with a given word length limitation. From 2008 the SOA-based cyberinfrastructure of LLS was re-used in Digital Humanities projects such as eAQUA and eTRACES (Büchler et al. 2008).

Results

Table 1 provides an overview of the 20 services offered with a breakdown of the requests and the responses. Over half of the requests (*64.6%*) were made to the *Baseform* service. Similarly, services with high-quality and often manually-curated data, such as the *Thesaurus* and *Synonyms* services, were requested more often than the quantitatively-computed *Similarity* service, which provided similarly used words by assuming the distributional hypothesis (Harris 1954), and thus compared the co-occurrence vectors of two words. Even if the coverage for this service, *66.02%*, is significantly higher than, for example, the *Category* (*35.92%*) or the *Synonyms* (*4.47%*) services, users appeared to prefer precision over recall for their end-user applications.



Service	*Requests*	*Requests (%)*	*Non-empty responses*	*Coverage (%)*	*Input Fields*	*Webservice Type*	*Access level*	*Installation date*
Baseform	624,275,884	64.636%	315,724,185	50.57%	W	MySQLSelect	FREE	04/2005
Category	120,476,452	12.473%	43,276,840	35.92%	W	MySQLSelect	FREE	04/2005
Thesaurus	69,573,648	7.203%	37,151,565	53.39%	W, L	MySQLSelect	FREE	04/2005
Synonyms	60,745,973	6.289%	2,719,544	4.47%	W, L	MySQLSelect	FREE	04/2005
Sentences	60,087,714	6.221%	11,536,172	19.19%	W, L	MySQLSelect	FREE	04/2005
Wordforms	12,671,302	1.311%	4,309,791	34.01%	W, L	MySQLSelect	FREE	04/2005
Frequencies	11,932,213	1.235%	8,095,420	67.84%	W	MySQLSelect	FREE	04/2005
LeftCollocationFinder	1,416,001	0.146%	295,714	20.88%	W, PoS, L	MySQLSelect	FREE	10/2005
RightCollocationFinder	1,379,356	0.142%	235,323	17.06%	W, PoS, L	MySQLSelect	FREE	10/2005
Cooccurrences	1,057,722	0.109%	629,795	59.54%	W, ST, L	MySQLSelect	FREE	04/2005
RightNeighbours	959,560	0.099%	567,870	59.18%	W, L	MySQLSelect	FREE	04/2005
LeftNeighbours	731,449	0.075%	473,600	64.74%	W, L	MySQLSelect	FREE	04/2005
Similarity	467,809	0.048%	308,877	66.02%	W, L	MySQLSelect	FREE	10/2005
CooccurrencesAll	20,852	0.002%	20,848	99.98%	W, ST, L	MySQLSelect	INTERN	05/2009
ExperimentalSynonyms	20,779	0.002%	14,860	71.51%	W, L	MySQLSelect	FREE	12/2009
Crossword puzzling	2,902	< 0.001%	1,306	45.00%	W, WL, L	MySQLSelect	FREE	10/2005
MARSService	616	< 0.001%	616	100.00%	W, L	MARS	INTERN	10/2006
NGrams	564	< 0.001%	149	26.41%	P, L	MySQLSelect	FREE	08/2011
NGramReferences	409	< 0.001%	87	21.27%	P, L	MySQLSelect	FREE	08/2011
Common co-occurrence	55	< 0.001%	43	78.18%	W1, W2, L	MySQLSelect	INTERN	10/2005
TOTAL	965,821,260		425,362,605					

Table 1. Overview of requests made to LLS between 2006-2014, in descending order. The Responses columns only list responses whose value was not empty. For space constraints, the values in the Input Fields column are abbreviated: Word (W.), Limit (L.), Pa

Low coverage is also caused by requests to German language databases, especially by compound nouns that cannot all be included in a *Baseform* or *Category* service. Many multi-word units (MWU) were also requested. Out of all the requests, *84,760,875* (*8.78%*) were MWUs. With regard to the distribution of the webservice usage, only the two most frequently requested services, *Baseform* and *Category*, were queried more often than the total count of the MWU requests. This speaks to the impact of MWUs.

The less frequently used webservices in table 1 were primarily limited to internal uses, to newly installed services or, as was the case for the Crossword Puzzling service, to manual usage instead of automatic bulk requests.

The following questions are discussed in the paper:

1. Geographical distribution and spread of requests
2. Requested languages distribution
3. Requests by cleanliness in terms of broken encodings or sending HTML code
4. Temporal distribution including lessons learnt from incompatibility issues of used software and their new versions causing a decrease in service usage
5. Identified service chains of the atomic LLS micro-services that users built on the client-side
6. Experiences for load balancing of linguistic services
7. Interoperability issues of programming languages and interpreting the WSDL-files differently
8. Comparisons of SOAP- and REST-based webservices

Conclusion

“If you build it, they will come“ is an infrastructure mantra that we can answer given the atomic micro-services of the LLS (more critical view by van Zundert 2012). However, with regard to easy-to-integrate and atomic micro-services we found that users were generally very pragmatic as they requested everything that they had found in texts or on webpages, such as RGB colour-sets, URLs and other meta-information. Based on the log-files, we conclude that it is easier to request a token and look for a match in the LLS database of millions of words rather than to invest only little time in conventional pre-processing and pre-selection on the client-side. Similarly, users repeatedly requested function words, sometimes only a few minutes apart. This user behaviour entailed a significant server load and user control over the requests. This type of recurring request on unchanged data could only be considered as spam.

We found that providing an infrastructure like the LLS over the course of a decade challenges the compatibility of used software components.

Moreover, from a Natural Language Processing (NLP) standpoint, the results contribute to existing conversations about the difficulty of building balanced and representative corpora. In fact, user interests detected in the LLS log-files can help to enrich corpora by adding further topics. The contribution also touches upon discussions about qualitative and manually-curated data versus automatically-computed and quantitatively-available results of language technology algorithms. Notwithstanding the improvement of NLP algorithms, our results show that users prefer qualitative data and that they often request these services even if the domain and concept coverage is relatively low. The conclusion we draw from the user behaviour observed in almost one billion requests is that research fields, including the Digital Humanities, should share their data –no matter how small– through large infrastructure initiatives like DARIAH and CLARIN in order to increase the textual coverage of linguistic resources.

Bibliography

Borchardt, S. (2005) *Generierbarkeit einer XML Topic Map aus E-Mails unter Verwendung von Text-Mining-Methoden und Nutzung von Web Services*. Bachelor thesis.

Büchler, M., Heyer, G., Gründer, S. (2008) *Bringing Modern Text Mining Approaches to Two Thousand Years Old Ancient Texts e-Humanities* At: Workshop in the 4th IEEE International Conference on e-Science.

Eckart, T., Quasthoff, U., and Goldhahn, D. (2012) *Language Statistics-Based Quality Assurance for Large Corpora*, Proceedings of Asia Pacific Corpus Linguistics Conference.

Harris, Z. (1954) Distributional structure, Word, 10, 2-3, pp. 146162.

Quasthoff, U., Richter, M., and Biemann, C. (2006) *Corpus Portal for Search in Monolingual Corpora* Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC).

Van Zundert, J. (2012) , *If You Build It, Will We Come? Large Scale Digital Infrastructures as a Dead End for Digital Humanities*, Historical Social Research / Historische Sozialforschung, vol. 37, no. 3, pp. 165-86.