Piotr Bański, Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder, Simon Clematide,
Marc Kupietz, Harald Lüngen, Caroline Iliadi

Proceedings of the Workshop on *Challenges in the Management of Large Corpora (CMLC-7) 2019*

Cardiff, 22 July 2019

IDS | LEIBNIZ-INSTITUT FÜR DEUTSCHE SPRACHE

IDS | LEIBNIZ-INSTITUT FÜR DEUTSCHE SPRACHE

Leibniz-Institut für Deutsche Sprache · R 5, 6-13 · 68161 Mannheim
www.ids-mannheim.de

URN: urn:nbn:de:bsz:mh39-89986
DOI: https://doi.org/10.14618/ids-pub-8998

Challenges in the Management of Large Corpora 2019

Workshop Programme 22 July 2019

Session 1 (09.00 - 11.00)

Johannes Graën, Tannon Kew, Anastassia Shaitarova and Martin Volk - "*Modelling Large Parallel Corpora*"

Pedro Javier Ortiz Suárez, Benoît Sagot and Laurent Romary - "*Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures"*

Vladimír Benko - *"Deduplication in Large Web Corpora"*

Mark Davies - *"The best of both worlds: Multi-billion word "dynamic" corpora"*

Session 2 (11.20 –13.00)

Andrew Hardie - *"Managing complex and arbitrary corpus subsections at scale and at speed: from formalism to implementation within CQPweb"*

Adrien Barbaresi - *"The Vast and the Focused: On the need for domain-focused web corpora"*

Marc Kupietz, Eliza Margaretha, Nils Diewald, Harald Lüngen and Peter Fankhauser - "*What's New in EuReCo? Interoperability, Comparable Corpora, Licensing"*

CMLC-7 Organising Committee

Piotr Bański, Marc Kupietz, Harald Lüngen	Leibniz-Institute for the German Language, Mannheim
Adrien Barbaresi	Berlin-Brandenburg Academy of Sciences
Hanno Biber, Evelyn Breiteneder	Austrian Academy of Sciences, Vienna
Simon Clematide	University of Zurich

CMLC-7 Programme Committee

Laurence Anthony	Waseda University, Japan
Vladimír Benko	Slovak Academy of Sciences
Felix Bildhauer	IDS Mannheim
Damir Ćavar	Indiana University, Bloomington
Mark Davies	BYU, USA
Tomaž Erjavec	Jožef Stefan Institute
Alexander Geyken	Berlin-Brandenburg Academy of Sciences and Humanities
Johannes Graën	University of Gothenburg, Pompeu Fabra University
Andrew Hardie	Lancaster University
Serge Heiden	ENS de Lyon
Miloš Jakubíček	Lexical Computing Ltd.
Michal Křen	Charles University, Prague
Sandra Kübler	Indiana University, Bloomington
Anke Lüdeling	HU Berlin
Piotr Pęzik	University of Łódź
Paul Rayson	Lancaster University
Martin Reynaert	Tilburg University
Laurent Romary	INRIA
Kevin Scannell	Saint-Louis University
Roland Schäfer	FU Berlin
Roman Schneider	Justus Liebig University Gießen, IDS Mannheim
Serge Sharoff	University of Leeds
Marko Tadić	University of Zagreb
Ludovic Tanguy	University of Toulouse
Dan Tufiş	Romanian Academy, Bucharest
Amir Zeldes	Georgetown University, USA

CMLC-7 Homepage: http://corpora.ids-mannheim.de/cmlc-2019.html

Table of contents

Author Index

Preface

The seventh CMLC meeting at CL2019 continued the successful series of "Challenges in the management of large corpora" events, previously hosted at LREC conferences, CL2015, and CL2017. As in the previous meetings, we aimed to explore common areas of interest across a range of issues in language resource management, corpus linguistics, natural language processing, and data science.

Large textual datasets require careful design, collection, cleaning, encoding, annotation, storage, retrieval, and curation to be of use for a wide range of research questions and to users across a number of disciplines. A growing number of national and other very large corpora are being made available, many historical archives are being digitised, numerous publishing houses are opening their textual assets for text mining, and many billions of words can be quickly sourced from the web and online social media.

A number of key themes and questions emerge that are of interest to the contributing research communities: (a) what can be done to deal with IPR and data protection issues? (b) what sampling techniques can we apply? (c) what quality issues should we be aware of? (d) what infrastructures and frameworks are being developed for the efficient storage, annotation, analysis and retrieval of large datasets? (e) what affordances do visualisation techniques offer for the exploratory analysis approaches of corpora? (f) what kinds of APIs or other means of access would make the corpus data as widely usable as possible without interfering with legal restrictions? (g) how to guarantee that corpus data remain available and usable in a sustainable way?

This year's event focused primarily on huge and complex datasets, across the entire spectrum of their life cycle: from the selection of data (including organizational and legal issues) and modelling of the eventual resources, through curation and all the way to analysis and visualisation. Attention was also paid to the ecosystem in which datasets thrive and interact – with interoperability being one of the meeting's leitmotifs.

We invite the readers to peruse the submissions collected in the present volume and to consider joining the CMLC community at our future meetings.

The CMLC-7 Organising Committee

Modelling Large Parallel Corpora: The Zurich Parallel Corpus Collection

Johannes Graën[1,2], **Tannon Kew**[3], **Anastassia Shaitarova**[3], **Martin Volk**[3]

[1]Department of Swedish, University of Gothenburg
[2]Department of Translation and Language Sciences, Pompeu Fabra University
[3]Institute of Computational Linguistics, University of Zurich

Abstract

Text corpora come in many different shapes and sizes and carry heterogeneous annotations, depending on their purpose and design. The true benefit of corpora is rooted in their annotation and the method by which this data is encoded is an important factor in their interoperability. We have accumulated a large collection of multilingual and parallel corpora and encoded it in a unified format which is compatible with a broad range of NLP tools and corpus linguistic applications. In this paper, we present our corpus collection and describe a data model and the extensions to the popular CoNLL-U format that enable us to encode it.

1 Introduction

The benefit of digital corpora is rooted in their annotation. In the history of corpus linguistics, several file formats have been employed to store and distribute digital corpora. Today, we see mainly two types of corpus formats that have prevailed: a tabular one, where each line represents a token and columns contain their attributes, and a hierarchical one, where tokens are represented as leaves of a tree.

Over the years, the Institute of Computational Linguistics in Zurich has accumulated a number of large parallel corpora in different languages that span various domains and genres, have multiple layers of annotation and carry rather heterogeneous metadata. So far, corpus data has generally been stored in XML files following an ad-hoc format that has never been fully standardised but adjusted to accommodate specific characteristics and annotation. In order to standardise our corpora and to make our data directly compatible with modern Natural Language Processing (NLP) tools, we extend the CoNLL-U format (Nivre et al. 2016). Since our corpora are parallel, or have large multiparallel parts, special attention is given to the representation of alignment information. Other types of annotation, including named entities and code switching are also accounted for.

This paper first describes the theoretical relational data model that we infer from over 10 years of work on the curation of corpora, the challenges faced and our considerations regarding compatibility and extensibility (Section 2). Then we propose an extended CoNLL-U format for storing parallel corpora with multiple layers of optional annotation (Section 3). This format facilitates the aggregation of data from different corpora while being directly compatible with relational databases, allowing for complex yet efficient queries. Lastly, we present our parallel corpus collection (Section 4), which is now made available in this standardised format.

2 Data Model

First we take a high-level view of our data and create a model which considers a compositional hierarchy of three entity types: tokens, sentences and texts. The token is typically the smallest unit in text corpora (but cf. Chiarcos et al. 2012), as such, annotation is predominantly performed on tokens on a sentence-by-sentence level.[1] In our corpora, sequences of tokens form sentences, although this may not be the case for all types of corpora (e.g. Bible verses (Christodouloupoulos and Steedman 2015) or subtitles (Lison and Tiedemann 2016) which may model verses or lines). Sentences often form paragraphs, which, in turn, form coherent texts.[2] While paragraphs typically subdivide texts into smaller thematic blocks, the concept of what constitutes a paragraph is somewhat arbitrary

[1]Exceptions are methods like coreference resolution or argument detection which require annotation across a sequence of sentences.

[2]We use 'text' to refer to a cohesive and coherent body of text within a corpus that could constitute a document, article or speaker turns in parliamentary debates.

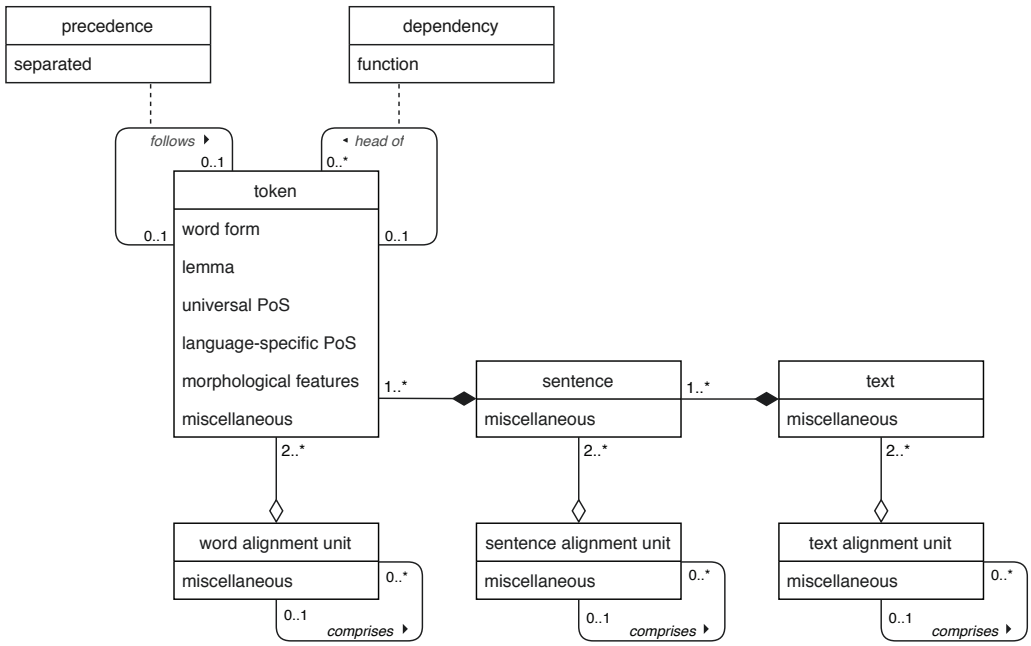


Figure 1: UML class diagram of a parallel corpus with potential hierarchical alignment on different levels.

and is often not consistently handled in different languages. Thus, we refrain from regarding paragraphs as an entity in our model, instead focusing on the hierarchy between tokens, sentences and texts (see Figure 1).

As most annotation in our corpora is centred around tokens, we model the token entity with common attributes such as surface form, lemma, part-of-speech tag and morphological features. Dependency grammar structures are represented through a recursive relationship between two tokens and defined by an attribute corresponding to the syntactic function. Here, an optional one-to-many cardinality describes dependency annotation suitable for tree structures. Graph structures can be expressed in a similar way if the source cardinality is loosened to allow for the representation of multiple heads for each token. The sequential order of tokens in a sentence is modelled as a precedence relation between two adjacent tokens. An attribute of this relation specifies whether tokens are separated by white space in the original surface form of a sentence, allowing for accurate reconstruction.

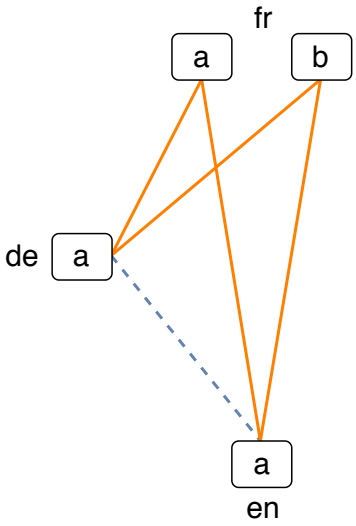
A 'miscellaneous' attribute at each level of the hierarchy allows for any relevant, unstructured information to be stored. For instance, to model both inter-sentential and intra-sentential code-switching (see Volk and Clematide 2014), we use this field to mark a token when its language deviates from that of its sentence and, similarly, for a sentence when its language differs from that of its text. While sentence and text entity types generally demand far fewer levels of annotation than tokens, the miscellaneous attribute permits arbitrary metadata, for example, formatting and layout information at the sentence level or speaker attribution at the text level.

2.1 Modelling Alignment

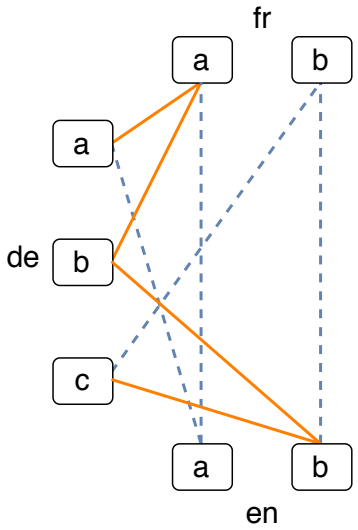
Alignment is modelled on token, sentence and text level as the affiliation of an entity to an alignment unit. This allows multilingual hierarchical alignment (Graën 2018, Sections 4.3 and 4.5) to be represented the same way as regular bilingual alignment. In most of our corpora, alignments are primarily bilingual.[3] In order to obtain multilingual alignments, we aggregate all corresponding bilingual alignments.[4] However, as illustrated in Figure 2, this approach does not always yield coherent and meaningful alignments across all languages. Figure 2a shows the ideal scenario, where the combination of one-to-one and one-to-many alignments is coherent, while in Figure 2b the combination results in an incoherent multilingual alignment. Nevertheless, modelling alignments in this way makes it possible to extract a subset of the available languages from any alignment unit.

[3] Except for the Sparcling corpus, which contains multilingual text and sentence alignments (Graën 2018).

[4] An alternative approach to representing multilingual alignment is to rely on a 'pivot' language (see Steinberger et al. 2014; Zeroual and Lakhouaja 2018).



(a) Coherence between one-to-one and one-to-many alignments.



(b) Incoherence between one-to-one and one-to-many alignments

Figure 2: Multiparallel alignment based on combining pairwise alignments with one-to-one relations (blue dashed edges) and one-to-many relations (orange solid edges).

3 Encoding Parallel Corpora

3.1 A smorgasbord of corpus formats

One of the most widely adopted approaches to encoding text corpora is XML (eXtensible Markup Language), which allows for a hierarchical representation using a tree structure. Such a representation is valuable for the storage of language data as it facilitates the clear separation of structural information from text content, provides a descriptive markup of the encoded text, and can easily be validated for consistency with an appropriate document type definition (DTD) or XML schema. For this reason, groups such as the Text Encoding Initiative[5] (TEI) have establish a standardised specification for the encoding of text corpora in XML (see also Dipper 2005; Hana and Štěpánek 2012; Gompel and Reynaert 2013).

A second approach is the tabular format that has quickly gained popularity and become the de facto standard in the NLP community (Buchholz and Marsi 2006; Chiarcos and Schenk 2018). The CoNLL-U format (Nivre et al. 2016) defines a standardised method of encoding text corpora for Universal Dependency (UD) Treebanks. It is based on a simple one-word-per-line (OWPL) format in which annotation layers are stored in ten distinct columns and are thus defined by their position, rather than markup tags.[6] This lightweight format is reminiscent of that used by the IMS Open Corpus Workbench (CWB) (Evert and the CWB Development Team 2010), which is able to blend both structural XML tags, albeit without being valid XML, and a tabular representation of a token's attributes in order to encode only the necessary linguistic information for a given task. Additionally, multiple extensions have been proposed to the basic CoNLL-U format, for example, for the annotation of multiword expressions (Savary et al. 2017) and morphological analysis (More et al. 2018). A more recent dialect of the CoNLL family is the CoNLL-U Plus format, which defines a modified CoNLL-U file that can contain any number of columns to flexibly encode any additional linguistic annotations while still maintaining a valid CoNLL format.

3.2 One format to rule them all

Despite the large number of corpus formats, there is little support for the representation of alignments. We decide to encode our corpora in what is essentially a CoNLL-U format and extend it with optional layers of stand-off annotation to accommodate the data model described in Section 2. Figure 3 depicts an excerpt from a corpus with multilingual alignments.

[5] https://tei-c.org/

[6] Numerous versions of the CoNLL format exist due to its application in multiple shared tasks held by the Conference on Computational Natural Language Learning (CoNLL) since 2006. CoNLL-U is an extension of CoNLL-X/CoNLL-ST which were themselves extensions of Joakim Nivre's MaltTAB format (Buchholz and Marsi 2006).

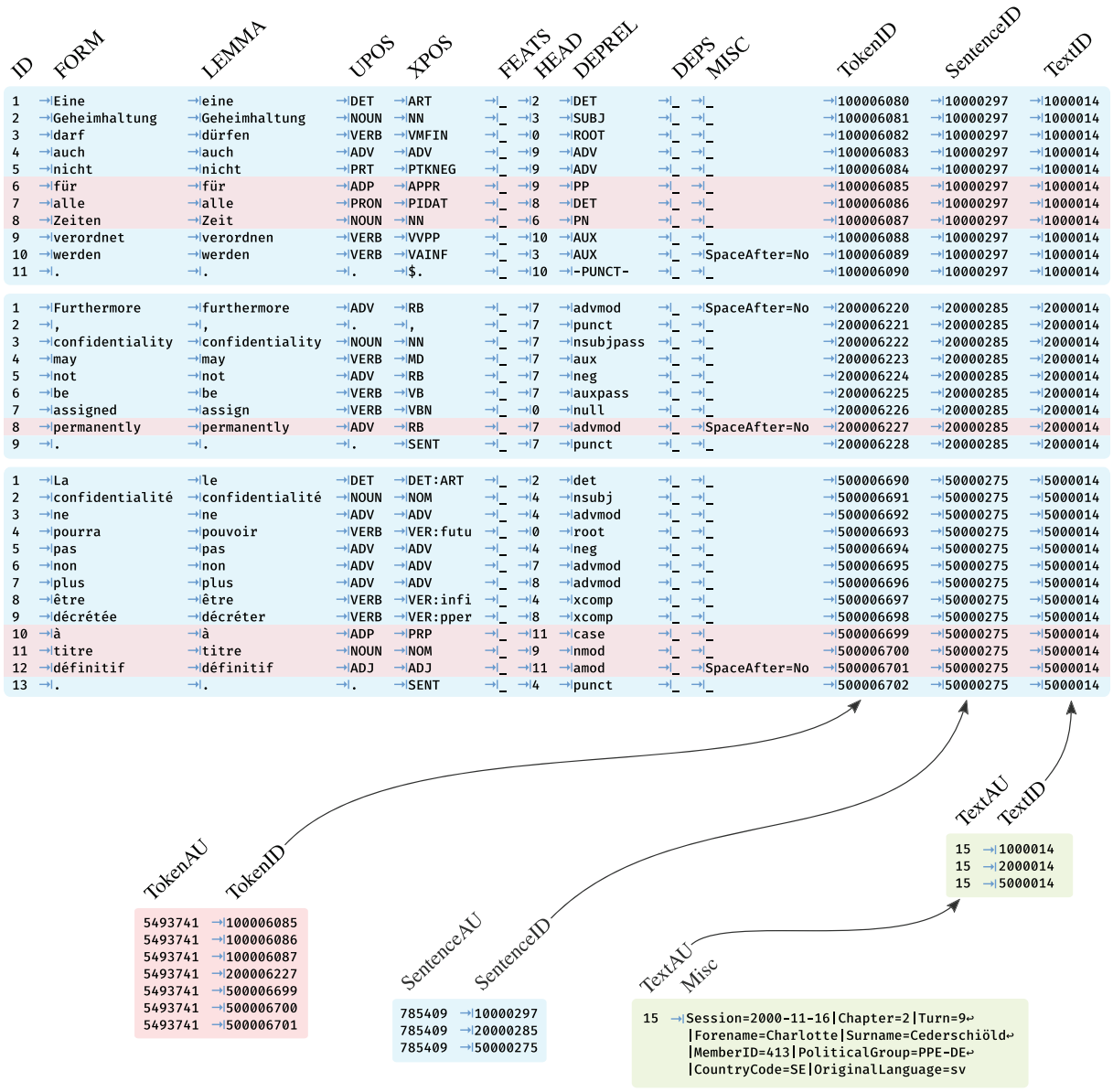


Figure 3: An excerpt of our extended CoNLL-U format for a parallel corpus with multilingual alignments. Snippets of stand-off files show token, sentence and text alignments. As depicted here, language-independent meta information can also be attached to alignment units.

Adopting CoNLL-U as a basis for our corpora brings a number of advantages: i) it ensures direct compatibility with numerous NLP tools[7], including state-of-the-art taggers and parsers, thereby making it easy to re-annotate our corpora as systems improve; ii) it guarantees that our corpora are directly compatible with relational database systems allowing for complex corpus queries; iii) it is human-readable and facilitates the extraction of language and task-specific data using simple command-line tools (e.g. `grep`, `sed`, `awk`); and iv) it provides a standardised base format for our large multilingual corpus collection, allowing for cross compatibility between corpora and serving as a good starting point for conversions into other transfer formats (e.g. TEI).

Naturally, there are some obvious shortcomings related to opting for a simplified tabular format to encode text corpora, some of which are discussed by Straňák and Štěpánek (2010) in their critique of the early CoNLL format. For example: i) multiple levels of sparse annotation can quickly lead to unwieldy tables; ii) corpus validation is made more difficult due to the lack of a DTD or schema for ensuring consistency; and iii) the inclusion of metadata and layout information, such

[7] https://universaldependencies.org/tools.html

as the placement of HTML tags, page breaks or graphics, which may be relevant for some analyses or veracity evaluation, is made difficult and cumbersome when moving away from XML markup.

3.3 Our Format

We split our corpora into language-specific subsections. For each section, tokens are furnished with ubiquitous attributes, pertaining to those specified by CoNLL-U in a main tabular token file.[8] These attributes include a sentence-positional identifier (word index), surface form, lemma, part-of-speech tags, morphological features, information for dependency relations and a miscellaneous attribute for additional token-level annotation. Unspecified or empty values are represented by an underscore ('_'). In the miscellaneous column, a list of attribute-value pairs is used to hold corpus-specific annotations at the token level (in the form of attribute=value, separated by pipe ('|') characters). In addition to the 10 columns defined by CoNLL-U, we include three enumerated identifier (ID) values. These IDs comprise one (primary) key, which uniquely identifies each token in a corpus, and two (foreign) keys, which reference the token's corresponding sentence and document.[9] All IDs are expected to increase linearly throughout the file, which facilitates processing.

Sentence-level and text-level annotations are then stored separately with relevant metadata based on their enumerated IDs. For consistency, we follow the same approach as in the token file and include a miscellaneous attribute for sentences and texts with a list of attribute-value pairs. Finally, we specify additional stand-off annotation files in order to accommodate non-ubiquitous annotation such as named entities and multilingual alignment. As such, stand-off files are only required when those annotations are present.

4 The Zurich Parallel Corpus Collection

Having brought our parallel corpus collection into a consistent and standardised format, as described in Section 3, we make these resources publicly available. This corpus collection provides a rich source of multilingual and multiparallel language data in a variety of domains and genres. A brief overview of the collection is given in Table 1.

At the heart of our collection lies the heritage corpus of alpine texts, **Text+Berg**[10] (Volk et al. 2010; Göhring and Volk 2011). This corpus consists of 150 years of digitised material from the Swiss Alpine Club yearbooks, which were published primarily in German and French, with some years containing texts in Italian, Romansh, English and also Swiss German.[11] Approximately 15% of the corpus comprises a German-French parallel subsection of roughly 4.5 million tokens per language. Over 10 years in development, Text+Berg has inspired numerous innovative approaches in corpus annotation, such as crowd-sourced correction of OCR errors (Clematide, Furrer et al. 2016), named entity recognition and linking (Ebling et al. 2011), code-switching (Volk and Clematide 2014), and special handling of elliptical compound nouns and separable prefix verbs in German (Volk, Clematide et al. 2016).

The **Credit Suisse Bulletin** corpus (CS Bulletin)[12] (Volk, Amrhein et al. 2016) is based on the world's oldest banking magazine published by Credit Suisse. This magazine has been in print since 1895 in both German and French, with translations also produced in English, Italian and Spanish at certain periods. There are more than 20 million tokens in the German and the French part, while the English and Italian sections contain about 10 million tokens per language. The Credit Suisse Bulletin corpus provides parallel data from magazine articles in the domains of economics, culture and sport, proving to be useful material for historic, sociological and linguistic research (Schneider et al. 2018).

The **Swiss Legislation Corpus** (SLC) (Höfler and Sugisaki 2014) is a German-French parallel corpus comprised of the entire classified collection of contemporary legislative writing of the Swiss Confederation. Its companion, the **Rumantsch Grischun corpus**[13] (Weibel 2014), consists of legal texts and press releases from the State Chancellery of the Swiss canton of Graubünden. This corpus provides unique parallel data for German and the low-resource language Romansh. As such, it is a valuable resource for Romansh language

[8] A header comment line beginning with '#' defines the columns and relevant namespaces, ensuring that it conforms with CoNLL-U Plus.

[9] Primary and foreign keys are terms borrowed from database design.

[10] http://textberg.ch/

[11] Although Swiss German has no official written standard, it is often written by native speakers in non-formal situations.

[12] https://pub.cl.uzh.ch/projects/b4c/en/

[13] 'Rumantsch' is an alternative spelling of 'Romansh'.

	languages	tokens	years	alignment
Text+Berg	de, fr, it, rm, gsw, en	52.6m	150	sentence
CS Bulletin	de, en, es, fr, it	61.6m	120	sentence
Sparcling	de, en, es, fr, it + 11	454.7m	15	token
SLC	de, fr	11.4m	—	token
Rumantsch Grischun	de, rm	0.9m	—	token
Medi-Notice	de, fr, it	58.9m	—	sentence
Horizons	de, en, fr	2.9m	14	text

Table 1: List of corpora together with their most relevant characteristics.

learners and a solid base for computational linguistic research.

The largest multiparallel corpus in our collection is the **Sparcling** corpus, originally referred to as FEP9 (Graën 2018). Sparcling is a richly annotated development of the CoStEP corpus (Graën et al. 2014), which itself is a cleaned and normalised version of the Europarl corpus (Koehn 2005). Token counts for each language vary, ranging from 7.5 to 47 million across the 16 languages, with annotation and alignment on all levels. Thus, it provides a rich resource for comparative language studies (Callegaro 2017), language learning applications (Schneider and Graën 2018) and the development of multilingual NLP methods (Heierli 2018). It has also been used in the implementation of a query and exploration system for multiparallel corpora (Clematide, Graën et al. 2016; Graën et al. 2017).

The **Medi-Notice** corpus (Fritz 2016) comprises texts from information leaflets for pharmaceutical products that are made publicly available by the Swiss Agency for Therapeutic Products. Each product usually has two separate leaflets: one is geared towards medical professionals, while the other is written for the general public. According to Swiss law, patient leaflets must be written in German, French and Italian, whereas the information for healthcare professionals is required only in German and French. Thus, the Medi-Notice corpus contains German and French parallel texts in the professional subsection, while the patient subsection is trilingual.

Lastly, the **Horizons** corpus[14] is a multiparallel corpus constructed from the magazine of the same name, published by the Swiss National Science Foundation.[15] This corpus also offers unique parallel texts in the domain of popular science in and around Switzerland in German, French and English.

5 Conclusions and Future Development

Through the development of the corpora mentioned above and the challenges involved in handling large multiparallel corpora, we have deduced a data model which allows us to represent the diversity of annotations in our corpora effectively. We have extended the CoNLL-U format to encode our corpora, which ensures compatibility with modern NLP applications and corpus linguistic tools, facilitates the extraction and the exploitation of linguistic data, and allows extensibility through various layers of stand-off annotation. Additionally, we have made our corpora available in this format, totalling approximately 640 million tokens across 18 languages. We hope that this will enable a more effective and efficient application of multiparallel corpora in a variety of linguistic research projects. At present, we are working on tools to handle corpora in our tabular format. This includes validation of the corpus files, extraction of task-specific subsections and conversion pipelines into other formats such as TEI. Further information and the corpus files are available at `https://pub.cl.uzh.ch/purl/PaCoCo`.

6 Acknowledgements

We would like to acknowledge the many contributors who have helped to develop the Zurich Parallel Corpus Collection described in this paper. Their extensive and valuable efforts over many years have made this current work possible. We would also like to thank the anonymous reviewers for their helpful comments and suggestions.

[14] The Horizons corpus has not yet been officially published and development is still underway, but it is being made available in its current form as part of this release.

[15] `https://www.horizons-mag.ch/`

References

Buchholz, Sabine and Erwin Marsi (2006). 'CoNLL-X Shared Task on Multilingual Dependency Parsing'. In: *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*. New York, pp. 149–164.

Callegaro, Elena (2017). 'Parallel Corpora for the Investigation of (Variable) Article Use in English: A Construction Grammar Approach'. PhD thesis. University of Zurich.

Chiarcos, Christian, Julia Ritz and Manfred Stede (2012). 'By all these lovely tokens… Merging Conflicting Tokenizations'. In: *Proceedings of the Linguistic Annotation Workshop (LAW)*, pp. 53–74.

Chiarcos, Christian and Niko Schenk (2018). 'The ACoLi CoNLL Libraries: Beyond Tab-Separated Values'. In: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari et al., pp. 571–576.

Christodouloupoulos, Christos and Mark Steedman (2015). 'A massively parallel corpus: the Bible in 100 languages'. In: *Language Resources and Evaluation* 49.2, pp. 375–395.

Clematide, Simon, Lenz Furrer and Martin Volk (2016). 'Crowdsourcing an OCR Gold Standard for a German and French Heritage Corpus'. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*. Portoroz, Slovenia, pp. 975–982.

Clematide, Simon, Johannes Graën and Martin Volk (2016). 'Multilingwis – A Multilingual Search Tool for Multi-Word Units in Multiparallel Corpora'. In: *Computerised and Corpus-based Approaches to Phraseology: Monolingual and Multilingual Perspectives – Fraseologia computacional y basada en corpus: perspectivas monolingües y multilingües*. Ed. by Gloria Corpas Pastor. Geneva: Tradulex, pp. 447–455.

Dipper, Stefanie (2005). 'XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation'. In: *Proceedings of Berliner XML Tage*, pp. 39–50.

Ebling, Sarah, Rico Sennrich, David Klaper and Martin Volk (2011). 'Digging for Names in the Mountains: Combined Person Name Recognition and Reference Resolution for German Alpine Texts'. In: *5th Language & Technology Conference (LTC)*, pp. 189–200.

Evert, Stefan and the CWB Development Team (2010). *The IMS Open Corpus Workbench (CWB) CQP Query Language Tutorial*.

Fritz, Andrea (2016). 'Erstellung eines parallelen Arzneimittelinformations-Korpus (Deutsch-Französisch) und Optimierung von dafür einsetzbaren Part-of-Speech-Taggern'. MA thesis. University of Zurich.

Göhring, Anne and Martin Volk (2011). 'The Text+Berg Corpus – An Alpine French-German Parallel Resource'. In: *Traitement Automatique des Langues Naturelles*, pp. 63–68.

Gompel, Maarten van and Martin Reynaert (2013). 'FoLiA: A practical XML format for linguistic annotation – a descriptive and comparative study'. In: *Computational Linguistics in the Netherlands* 3, pp. 63–81.

Graën, Johannes (2018). 'Exploiting Alignment in Multiparallel Corpora for Applications in Linguistics and Language Learning'. PhD thesis. University of Zurich.

Graën, Johannes, Dolores Batinic and Martin Volk (2014). 'Cleaning the Europarl Corpus for Linguistic Applications'. In: *Proceedings of the 12th Conference on Natural Language Processing (KONVENS)*, pp. 222–227.

Graën, Johannes, Dominique Sandoz and Martin Volk (2017). 'Multilingwis2 – Explore Your Parallel Corpus'. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*. Linköping Electronic Conference Proceedings 131, pp. 247–250.

Hana, Jirka and Jan Štěpánek (2012). 'Prague Markup Language Framework'. In: *Proceedings of the 6th Linguistic Annotation Workshop (LAW)*. Jeju, Republic of Korea, pp. 12–21.

Heierli, Jasmin (2018). 'Lemma Disambigation in Multilingual Parallel Corpora'. MA thesis. University of Zurich.

Höfler, Stefan and Kyoko Sugisaki (2014). 'Constructing and Exploiting an Automatically Annotated Resource of Legislative Texts'. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari et al., pp. 175–180.

Koehn, Philipp (2005). 'Europarl: A parallel corpus for statistical machine translation'. In: *Proceedings of the 10th Machine Translation Sum-*

mit. Vol. 5. Asia-Pacific Association for Machine Translation (AAMT), pp. 79–86.

Lison, Pierre and Jörg Tiedemann (2016). 'OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles'. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari et al., pp. 923–929.

More Amirand Çetinoğlu, Özlem, Çağrı Çöltekin, Nizar Habash, Benoît Sagot, Djamé Seddah, Dima Taji and Reut Tsarfaty (2018). 'CoNLL-UL: Universal Morphological Lattices for Universal Dependency Parsing'. In: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari et al., pp. 3847–3853.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty and Daniel Zeman (2016). 'Universal Dependencies v1: A Multilingual Treebank Collection'. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari et al., pp. 1659–1666.

Savary, Agata, Carlos Ramisch, Silvio Cordeiro, Federico Sangati and Veronika Vincze (2017). 'The PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions'. In: *Proceedings of the 13th Workshop on Multiword Expressions*. Valencia, Spain, pp. 31–47.

Schneider, Gerold and Johannes Graën (2018). 'NLP Corpus Observatory – Looking for Constellations in Parallel Corpora to Improve Learners' Collocational Skills'. In: *Proceedings of the 7th workshop on NLP for Computer Assisted Language Learning (NLP4CALL)*, pp. 69–78.

Schneider, Gerold, Anastassia Shaitarova and Martin Volk (2018). 'Credit Suisse Bulletin Corpus: The world's Oldest Banking Magazine as a Treasure Trove of Applications for Digital Humanities'. Poster at Workshop DARIAH-CH. University of Neuchâtel.

Steinberger, Ralf, Mohamed Ebrahim, Alexandros Poulis, Manuel Carrasco-Benitez, Patrick Schlüter, Marek Przybyszewski and Signe Gilbro (2014). 'An overview of the European Union's highly multilingual parallel corpora'. In: *Language Resources and Evaluation* 48.4, pp. 679–707.

Straňák, Pavel and Jan Štěpánek (2010). 'Representing Layered and Structured Data in the CoNLL-ST Format'. In: *Proceedings of the 2nd International Conference on Global Interoperability for Language Resources (ICGL)*, pp. 143–152.

Volk, Martin, Chantal Amrhein, Noëmi Aepli, Mathias Müller and Phillip Ströbel (2016). 'Building a Parallel Corpus on the World's Oldest Banking Magazine'. In: *Proceedings of the 13th Conference on Natural Language Processing (KONVENS)*, pp. 288–296.

Volk, Martin, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer and Beni Ruef (2010). 'Challenges in Building a Multilingual Alpine Heritage Corpus'. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari et al.

Volk, Martin and Simon Clematide (2014). 'Detecting Code-Switching in a Multilingual Alpine Heritage Corpus'. In: *Proceedings of the 1st Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 24–33.

Volk, Martin, Simon Clematide, Johannes Graën and Phillip Ströbel (2016). 'Bi-particle Adverbs, PoS-Tagging and the Recognition of German Separable Prefix Verbs'. In: *Proceedings of the 13th Conference on Natural Language Processing (KONVENS)*, pp. 297–305.

Weibel, Manuela (2014). 'Aufbau paralleler Korpora und Implementierung eines wortalignierten Suchsysteme für Deutsch – Rumantsch Grischun'. MA thesis. University of Zurich.

Zeroual, Imad and Abdelhak Lakhouaja (2018). 'MulTed: A Multilingual Aligned and Tagged Parallel Corpus'. In: *Applied Computing and Informatics*.

Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures

Pedro Javier Ortiz Suárez[1,2] **Benoît Sagot**[1] **Laurent Romary**[1]
[1]Inria, Paris, France
[2]Sorbonne Université, Paris, France
{pedro.ortiz, benoit.sagot, laurent.romary}@inria.fr

Abstract

Common Crawl is a considerably large, heterogeneous multilingual corpus comprised of crawled documents from the internet, surpassing 20TB of data and distributed as a set of more than 50 thousand plain text files where each contains many documents written in a wide variety of languages. Even though each document has a metadata block associated to it, this data lacks any information about the language in which each document is written, making it extremely difficult to use Common Crawl for monolingual applications. We propose a general, highly parallel, multithreaded pipeline to clean and classify Common Crawl by language; we specifically design it so that it runs efficiently on medium to low resource infrastructures where I/O speeds are the main constraint. We develop the pipeline so that it can be easily reapplied to any kind of heterogeneous corpus and so that it can be parameterised to a wide range of infrastructures. We also distribute a 6.3TB version of Common Crawl, filtered, classified by language, shuffled at line level in order to avoid copyright issues, and ready to be used for NLP applications.

1 Introduction

In recent years neural methods for Natural Language Processing (NLP) have consistently and repeatedly improved the state-of-the-art in a wide variety of NLP tasks such as parsing, PoS-tagging, named entity recognition, machine translation, text classification and reading comprehension among others. Probably the main contributing factor in this steady improvement for NLP models is the raise in usage of *transfer learning* techniques in the field. These methods normally consist of taking a pre-trained model and reusing it, with little to no retraining, to solve a different task from the original one it was intended to solve; in other words, one *transfers* the *knowledge* from one task to another.

Most of the transfer learning done in NLP nowadays is done in an unsupervised manner, that is, it normally consist of a *language model* that is fed unannotated plain text in a particular language; so that it *extracts* or *learns* the basic *features* and patterns of the given language, the model is subsequently used on top of an specialised architecture designed to tackle a particular NLP task. Probably the best known example of this type of model are *word embeddings* which consist of real-valued vector representations that are trained for each word on a given corpus. Some notorious examples of word embeddings are word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and fastText (Mikolov et al., 2018). All these models are *context-free*, meaning that a given word has one single vector representation that is independent of context, thus for a polysemous word like Washington, one would have one single representation that is reused for the city, the state and the US president.

In order to overcome the problem of polysemy, *contextual* models have recently appeared. Most notably ELMo (Peters et al., 2018) which produces deep contextualised word representations out of the internal states of a deep bidirectional language model in order to model word use and how the usage varies across linguistic contexts. ELMo still needs to be used alongside a specialised architecture for each given downstream task, but newer architectures that can be fine-tuned have also appear. For these, the model is first fed unannotated data, and is then fine-tuned with annotated data to a particular downstream task without relying on any other architecture. The most remarkable examples of this type of model are GPT-1, GPT-2 (Radford et al., 2018, 2019), BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019);

the latter being the current state-of-the-art for multiple downstream tasks. All of these models are different arrangements of the Transformer architecture (Vaswani et al., 2017) trained with different datasets, except for XLNet which is an instance of the Transformer-XL (Dai et al., 2019).

Even though these models have clear advantages, their main drawback is the amount of data that is needed to train them in order to obtain a functional and efficient model. For the first English version of word2vec, Mikolov et al. (2013) used a one billion word dataset consisting of various news articles. Later Al-Rfou et al. (2013) and then Bojanowski et al. (2017) used the plain text from Wikipedia to train distributions of word2vec and fastText respectively, for languages other than English. Now, the problem of obtaining large quantities of data aggravates even more for contextual models, as they normally need multiple instances of a given word in order to capture all its different uses and in order to avoid overfitting due to the large quantity of hyperparameters that these models have. Peters et al. (2018) for example use a 5.5 billion token[1] dataset comprised of crawled news articles plus the English Wikipedia in order to train ELMo, Devlin et al. (2018) use a 3.3 billion word[2] corpus made by merging the English Wikipedia with the BooksCorpus (Zhu et al., 2015), and Radford et al. (2019) use a 40GB English corpus created by scraping outbound links from Reddit.[3]

While Wikipedia is freely available, and multiple pipelines exist[4,5] to extract plain text from it, some of the bigger corpora mentioned above are not made available by the authors either due to copyright issues or probably because of the infrastructure needed to serve and distribute such big corpora. Moreover the vast majority of both these models and the corpora they are trained with are in English, meaning that the availability of high quality NLP for other languages, specially for low-resource languages, is rather limited.

To address this problem, we choose Common Crawl,[6] which is a 20TB mutilingual free to use corpus composed of crawled websites from the internet, and we propose a highly parallel multi-threaded asynchronous pipeline that applies well-known concurrency patterns, to clean and classify by language the whole Common Crawl corpus to a point where it is usable for Machine Learning and in particular for neural NLP applications. We optimise the pipeline so that the process can be completed in a sensible amount of time even in infrastructures where Input/Output (I/O) speeds become the main bottleneck.

Knowing that even running our pipeline will not always be feasible, we also commit to publishing our own version of a classified by language, filtered and ready to use Common Crawl corpus upon publication of this article. We will set up an easy to use interface so that people can download a manageable amount of data on a desired target language.

2 Related Work

Common Crawl has already been successfully used to train language models, even multilingual ones. The most notable example in probably fastText which was first trained for English using Common Crawl (Mikolov et al., 2018) and then for other 157 different languages (Grave et al., 2018). In fact Grave et al. (2018) proposed a pipeline to filter, clean and classify their fastText multilingual word embeddings, which we shall call the "fastText pre-processing pipeline." They used the fastText linear classifier (Joulin et al., 2016, 2017) to classify each line of Common Crawl by language, and downloaded the initial corpus and schedule the I/O using some simple Bash scripts. Their solution, however, proved to be a synchronous blocking pipeline that works well on infrastructures having the necessary hardware to assure high I/O speeds even when storing tens of terabytes of data at a time. But that downscales poorly to medium-low resource infrastructures that rely on more traditional cost-effective electromechanical mediums in order to store this amount of data.

Concerning contextual models, Baevski et al. (2019) trained a BERT-like bi-directional Transformer for English using Common Crawl. They followed the "fastText pre-processing pipeline" but they removed all copies of Wikipedia inside Common Crawl. They also trained their model using News Crawl (Bojar et al., 2018) and using Wikipedia + BooksCorpus, they compared three

[1] Punctuation marks are counted as tokens.
[2] Space sparated tokens.
[3] https://www.reddit.com/
[4] https://github.com/attardi/wikiextractor
[5] https://github.com/hghodrati/wikifil
[6] http://commoncrawl.org/

models and showed that Common Crawl gives the best performance out of the three corpora.

The XLNet model was trained for English by joining the BookCorpus, English Wikipedia, Giga5 (Parker et al., 2011), ClueWeb 2012-B (Callan et al., 2009) and Common Crawl. Particularly for Common Crawl, Yang et al. (2019) say they use "heuristics to aggressively filter out short or low-quality articles" from Common Crawl, however they don't give any detail about these "heuristics" nor about the pipeline they use to classify and extract the English part of Common Crawl.

It is important to note that none of these projects distributed their classified, filtered and cleaned versions of Common Crawl, making it difficult in general to faithfully reproduce their results.

3 Common Crawl

Common Crawl is a non-profit foundation which produces and maintains an open repository of web crawled data that is both accessible and analysable.[7] Common Crawl's complete web archive consists of petabytes of data collected over 8 years of web crawling. The repository contains raw web page HTML data (WARC files), metdata extracts (WAT files) and plain text extracts (WET files). The organisation's crawlers has always respected `nofollow`[8] and `robots.txt`[9] policies.

Each monthly Common Crawl snapshot is in itself a massive multilingual corpus, where every single file contains data coming from multiple web pages written in a large variety of languages and covering all possible types of topics. Thus, in order to effectively use this corpus for the previously mentioned Natural Language Processing and Machine Learning applications, one has first to extract, filter, clean and classify the data in the snapshot by language.

For our purposes we use the WET files which contain the extracted plain texts from the websites mostly converted to UTF-8, as well as headers containing the metatada of each crawled document. Each WET file comes compressed in gzip format[10] and is stored on Amazon Web Services. We use the November 2018 snapshot which surpasses 20TB of uncompressed data and contains more than 50 thousand plain text files where each file consists of the plain text from multiple websites along its metadata header. From now on, when we mention the "Common Crawl" corpus, we refer to this particular November 2018 snapshot.

4 fastText's Pipeline

In order to download, extract, filter, clean and classify Common Crawl we base ourselves on the "fastText pre-processing pipeline" used by Grave et al. (2018). Their pipeline first launches multiple process, preferably as many as available cores. Each of these processes first downloads one Common Crawl WET file which then proceeds to decompress after the download is over. After decompressing, an instance of the fastText linear classifier (Joulin et al., 2016, 2017) is launched, the classifier processes each WET file line by line, generating a language tag for each line. The tags are then stored in a tag file which holds a one-to-one correspondence between lines of the WET file and its corresponding language tag. The WET file and the tag files are read sequentially and each on the WET file line holding the condition of being longer that 100 bytes is appended to a language file containing only plain text (tags are discarded). Finally the tag file and the WET files are deleted.

Only when one of these processes finishes another can be launched. This means that one can at most process and download as many files as cores the machine has. That is, if for example a machine has 24 cores, only 24 WET files can be downloaded and processed simultaneously, moreover, the 25$^{\text{th}}$ file won't be downloaded until one of the previous 24 files is completely processed.

When all the WET files are classified, one would normally get around 160 language files, each file holding just plain text written in its corresponding language. These files still need to be filtered in order to get rid of all files containing invalid UTF-8 characters, so again a number of processes are launched, this time depending on the amount of memory of the machine. Each process reads a language file, first filters for invalid UTF-8 characters and then performs deduplication. A simple non-collision resistant hashing algorithm is used to deduplicate the files.

The fastText linear classifier works by repre-

[7] http://commoncrawl.org/about/

[8] http://microformats.org/wiki/rel-nofollow

[9] https://www.robotstxt.org/

[10] https://www.gnu.org/software/gzip/

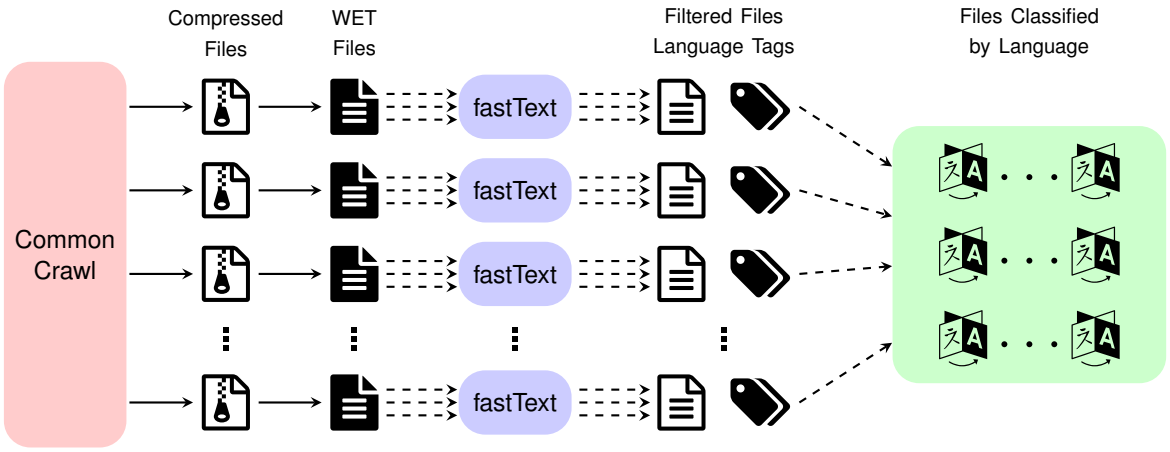


Figure 1: A scheme of the *goclassy* pipeline. The red square represents the Compressed WET files stored on Amazon Web Services. The icons represent the gzip files stored locally, the represent one of the 50K WET files. The represents the filtered file and the represents a file of language tags, one tag per line in . The represents one of the 166 classified files. Each arrow represents an asynchronous non blocking worker and dotted arrows represent a line filtering process.

senting sentences for classification as Bags of Words (BoW) and training a linear classifier. A weight matrix A is used as a look-up table over the words and the word representations are then averaged into a text representation which is fed to the linear classifier. The architecture is in general similar to the CBoW model of Mikolov et al. (2013) but the middle word is replaced by a label. They uses a softmax function f to compute the probability distribution over the classes. For a set of N documents, the model is trained to minimise the negative log-likelihood over the classes:

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log\left(f(BAx_n)\right),$$

where x_n is the normalised bag of features of the n-th document, y_n is the n-th label, and A, B are the weight matrices. The pre-trained fastText model for language recognition (Grave et al., 2018) is capable of recognising around 176 different languages and was trained using 400 million tokens from Wikipedia as well as sentences from the Tatoeba website[11].

5 Asynchronous pipeline

We propose a new pipeline derived from the fastText one which we call `goclassy`, we reuse the fastText linear classifier (Joulin et al., 2016, 2017) and the pre-trained fastText model for language recognition (Grave et al., 2018), but we completely rewrite and parallelise their pipeline in an asynchronous manner.

The order of operations is more or less the same as in the fastText pre-processing pipeline but instead of clustering multiple operations into a single blocking process, we launch a worker for each operation and we bound the number of possible parallel operations at a given time by the number of available threads instead of the number of CPUs. We implement goclassy using the Go programming language[12] so we let the Go runtime[13] handle the scheduling of the processes. Thus in our pipeline we don't have to wait for a whole WET file to download, decompress and classify in order to start downloading and processing the next one, a new file will start downloading and processing as soon as the scheduler is able to allocate a new process.

When using electromechanical mediums of storage, I/O blocking is one of the main problems one encounters. To overcome this, we introduced buffers in all our I/O operations, a feature that is not present in the fastText pre-processing pipeline. We also create, from the start, a file for each of the 176 languages that the pre-trained fastText language classifier is capable of recognising, and we always leave them open, as we find that getting a file descriptor to each time we want to write, if we wanted leave them open just when needed, introduces a big overhead.

[11] https://tatoeba.org/

[12] https://golang.org/

[13] https://golang.org/src/runtime/mprof.go

	10 files			100 files			200 files		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
real									
fastText	2m50s	6m45s	3m31s	13m46s	38m38s	17m39s	26m20s	47m48s	31m4s
goclassy	1m23s	3m12s	1m42s	7m42s	12m43s	9m8s	15m3s	15m47s	15m16s
user									
fastText	26m45s	27m2s	26m53s	4h21m	4h24m	4h23m	8h42m	8h48m	8h45m
goclassy	10m26s	12m53s	11m0s	1h46m	1h54m	1h49m	3h37m	3h40m	3h38m
sys									
fastText	40.14s	40.85s	40.56s	6m14s	6m17s	6m15s	12m26s	12m45s	12m31s
goclassy	37.34s	45.98s	39.67s	5m7s	5m34s	5m16s	9m57s	10m14s	10m5s

Table 1: Benchmarks are done using the UNIX `time` tool, are repeated 10 times each and are done for random samples of 10, 100 and 200 WET files. Only the classifying and filtering part are benchmarked. The table shows the minimum, maximum and mean time for the user, real and sys time over the 10 runs. Here "fastText" is used as short for the pipeline.

We also do the filtering and cleaning processes at line level before feeding each line to the classifier, which makes us create a new filtered file so that we can have a correspondence with the tag file, which in turn will consume more space, but that will also reduce the amount of unnecessary classifications performed by fastText. The filtered and file tags are then read and lines are appended to its corresponding language file. The writing in the classification step is asynchronous, meaning that process writing a line to the filtered files does not wait for the classifier to write a tag on the tag file. Figure 1 shows the pipeline up to this point.

After all WET files are processed, we then use Isaac Whitfield's deduplication tool runiq[14] which is based on Yann Collet's xxhash64[15], an extremely fast non-cryptographic hash algorithm that is resistant to collisions. We finally use the Mark Adler's pigz[16] for data compression, as opposed to the canonical UNIX tools proposed in the original fastText pipeline. We add both tools to our concurrent pipeline, executing multiple instances of them in parallel, in order to ensure we use the most of our available resources at a given time.

Beyond improving the computational time required to classify this corpus, we propose a simple improvement on the cleaning scheme in the fastText pre-processing pipeline. This improvement allows our pipeline to better take into account the multilingual nature of Common Crawl; that is, we count UTF-8 characters instead of bytes for setting the lower admissible bound for the length of a line to be fed into the classifier. This straightforward modification on the fastText pre-processing pipeline assures we take into account the multiple languages present in Common Crawl that use non-ASCII encoded characters.

Given that our implementation is written in Go, we release binary distributions [17] of goclassy for all major operating systems. Both pigz and runiq are also available for all major operating systems.

6 Benchmarks

We test both pipelines against one another in an infrastructure using traditional electromechanical storage mediums that are connected to the main processing machine via an Ethernet interface, that is, a low I/O speed environment as compared to an infrastructure where one would have an array of SSDs connected directly to the main processing machine via a high speed interface. We use a machine with an Intel® Xeon® Processor E5-2650 2.00 GHz, 20M Cache, and 203.1 GiB of RAM. We make sure that no other processes apart from the benchmark and the Linux system processes are run. We do not include downloading, decompression or deduplication in our benchmarks as downloading takes far too much time, and deduplication and compression were performed with third party tools that don't make part of our main contribution. We are mainly interested in seeing how the way the data is fed to the classifier impacts the overall processing time.

Benchmarks in table 1 of our goclassy pipeline show a drastic reduction in processing time compared to the original fastText prepossessing pipeline. We show that in our particular infrastructure, we are capable of reducing the *real* time

[14] https://github.com/whitfin/runiq
[15] https://github.com/Cyan4973/xxHash
[16] https://zlib.net/pigz/
[17] https://github.com/pjox/goclassy

as measured by the `time` UNIX tool almost always by half. The *user* time which represents the amount of CPU time spent in user-mode code (outside the kernel) within the process is almost three times lower for our goclassy pipeline, this particular benchmark strongly suggest a substantial reduction in energy consumption of goclassy with respect to the fastText pipeline.

As we understand that even an infrastructure with more than 20TB of free space in traditional electromechanical storage is not available to everyone and we propose a simple parametrization in our pipeline that actively deletes already processed data and that only downloads and decompresses files when needed, thus ensuring that no more than 10TB of storage are used at a given time. We nevertheless note that delaying decompression increases the amount of computation time, which is a trade-off that some users might make as it might be more suitable for their available infrastructure.

7 OSCAR

Finally, we are aware that some users might not even have access to a big enough infrastructure to run our pipelines or just to store all the Common Crawl data. Moreover, even if previously used and cited in NLP and Machine Learning research, we note that there is currently no public distribution of Common Crawl that is filtered, classified by language and ready to use for Machine Learning or NLP applications. Thus we decide to publish a pre-processed version of the November 2018 copy of Common Crawl which is comprised of usable data in 166 different languages, we publish[18] our version under the name OSCAR which is short for *Open Super-large Crawled ALMAnaCH*[19] *coRpus*.

After processing all the data with goclassy, the size of the whole Common Crawl corpus is reduced to 6.3TB, but in spite of this considerable reduction, OSCAR still dwarfs all previous mentioned corpora having more 800 billion "words" or spaced separated tokens and noting that this in fact in an understatement of how big OSCAR is, as some of the largest languages within OSCAR such as Chinese and Japanese do not use spaces. The sizes in bytes for both the original and the deduplicated versions of OSCAR can be found in table 2. OSCAR is published under the *Creative Commons CC0 license ("no rights reserved")*[20], so it is free to use for all applications.

8 Conclusions

We are sure that our work will greatly benefit researchers working on an either constrain infrastructure or a low budget setting. We are also confident, that by publishing a classified version of Common Crawl, we will substantially increase the amount of available public data for medium to low resource languages, thus improving and facilitating NLP research for them. Furthermore, as our pipeline speeds-up and simplifies the treatment of Common Crawl, we believe that our contribution can be further parallelised and adapted to treat multiple snapshots of Common Crawl opening the door to what would be otherwise costly diachronic studies of the use of a given language throughout the internet.

Finally, we note that both our proposed pipeline is data independent, which means that they can be reused to process, clean and classify any sort of big multilingual corpus that is available in plain text form and that is UTF-8 encoded; meaning that the impact of our work goes way beyond a single corpus.

Acknowledgements

The authors are grateful to Inria Paris *"rioc"* computation cluster for providing resources and support, and for allowing us to store the complete copies of both the raw and the filtered Common Crawl versions on their infrastructure.

[18] https://team.inria.fr/almanach/oscar/
[19] https://team.inria.fr/almanach/
[20] http://creativecommons.org/publicdomain/zero/1.0/

Language	Size		Words		Language	Size		Words	
	Orig	Dedup	Orig	Dedup		Orig	Dedup	Orig	Dedup
Afrikaans	241M	163M	43,482,801	29,533,437	Lower Sorbian	13K	7.1K	1,787	966
Albanian	2.3G	1.2G	374,196,110	186,856,699	Luxembourgish	29M	21M	4,403,577	3,087,650
Amharic	360M	206M	28,301,601	16,086,628	Macedonian	2.1G	1.2G	189,289,873	102,849,595
Arabic	82G	32G	8,117,162,828	3,171,221,354	Maithili	317K	11K	69,161	874
Aragonese	1.3M	801K	52,896	45,669	Malagasy	21M	13M	3,068,360	1,872,044
Armenian	3.7G	1.5G	273,919,388	110,196,043	Malay	111M	42M	16,696,882	6,045,753
Assamese	113M	71M	6,956,663	4,366,570	Malayalam	4.9G	2.5G	189,534,472	95,892,551
Asturian	2.4M	2.0M	381,005	325,237	Maltese	24M	17M	2,995,654	2,163,358
Avaric	409K	324K	24,720	19,478	Marathi	2.7G	1.4G	162,609,404	82,130,803
Azerbaijani	2.8G	1.5G	322,641,710	167,742,296	Mazanderani	691K	602K	73,870	64,481
Bashkir	128M	90M	9,796,764	6,922,589	Minangkabau	608K	310K	5,682	4,825
Basque	848M	342M	120,456,652	45,359,710	Mingrelian	5.8M	4.4M	299,098	228,629
Bavarian	503	503	399	399	Mirandese	1.2K	1.1K	171	152
Belarusian	1.8G	1.1G	144,579,630	83,499,037	Modern Greek	62G	27G	5,479,180,137	2,412,419,435
Bengali	11G	5.8G	623,575,733	363,766,143	Mongolian	2.2G	838M	181,307,167	68,362,013
Bihari	110K	34K	8,848	2,875	Nahuatl languages	12K	11K	1,234	1,193
Bishnupriya	4.1M	1.7M	198,286	96,940	Neapolitan	17K	13K	5,282	4,147
Bosnian	447K	116K	106,448	20,485	Nepali	1.8G	1.2G	107,448,208	71,628,317
Breton	29M	16M	5,013,241	2,890,384	Newari	5.5M	4.1M	564,697	288,995
Bulgarian	32G	14G	2,947,648,106	1,268,114,977	Northern Frisian	4.4K	4.4K	1,516	1,516
Burmese	1.9G	1.1G	56,111,184	30,102,173	Northern Luri	76K	63K	8,022	6,740
Catalan	8.0G	4.3G	1,360,212,450	729,333,440	Norwegian	8.0G	4.7G	1,344,326,388	804,894,377
Cebuano	39M	24M	6,603,567	3,675,024	Norwegian Nynorsk	85M	54M	14,764,980	9,435,139
Central Bikol	885	885	312	312	Occitan	5.8M	3.7M	750,301	512,678
Central Khmer	1.1G	581M	20,690,610	10,082,245	Oriya	248M	188M	14,938,567	11,321,740
Central Kurdish	487M	226M	48,478,334	18,726,721	Ossetian	13M	11M	1,031,268	878,765
Chavacano	520	520	130	130	Pampanga	760	304	130	52
Chechen	8.3M	6.7M	711,051	568,146	Panjabi	763M	460M	61,847,806	37,555,835
Chinese	508G	249G	14,986,424,850	6,350,215,113	Persian	79G	38G	9,096,554,121	4,363,505,319
Chuvash	39M	26M	3,041,614	2,054,810	Piemontese	2.1M	1.9M	362,013	337,246
Cornish	44K	14K	8,329	2,704	Polish	109G	47G	15,277,255,137	6,708,709,674
Croatian	226M	110M	34,232,765	16,727,640	Portuguese	124G	64G	20,641,903,898	10,751,156,918
Czech	53G	24G	7,715,977,441	3,540,997,509	Pushto	361M	242M	46,559,441	31,347,348
Danish	16G	9.5G	2,637,463,889	1,620,091,317	Quechua	78K	67K	10,186	8,691
Dhivehi	126M	79M	7,559,472	4,726,660	Romanian	25G	11G	3,984,317,058	1,741,794,069
Dimli	146	146	19	19	Romansh	7.4K	6.5K	1,093	960
Dutch	78G	39G	13,020,136,373	6,598,786,137	Russia Buriat	13K	11K	963	809
Eastern Mari	7.2M	6.0M	565,992	469,297	Russian	1.2T	568G	92,522,407,837	46,692,691,520
Egyptian Arabic	66M	33M	7,305,151	3,659,419	Sanskrit	93M	37M	4,331,569	1,713,930
Emilian-Romagnol	25K	24K	6,376	6,121	Scottish Gaelic	1.9M	1.3M	310,689	207,110
English	2.3T	1.2T	418,187,793,408	215,841,256,971	Serbian	3.9G	2.2G	364,395,411	207,561,168
Erzya	1.4K	1.2K	90	78	Serbo-Croatian	25M	5.8M	5,292,184	1,040,573
Esperanto	299M	228M	48,486,161	37,324,446	Sicilian	3.3K	2.8K	554	468
Estonian	4.8G	2.3G	643,163,730	309,931,463	Sindhi	347M	263M	43,530,158	33,028,015
Finnish	27G	13G	3,196,666,419	1,597,855,468	Sinhala	1.4G	802M	93,053,465	50,864,857
French	282G	138G	46,896,036,417	23,206,776,649	Slovak	9.1G	4.5G	1,322,247,763	656,346,179
Galician	620M	384M	102,011,291	63,600,602	Slovenian	2.5G	1.3G	387,399,700	193,926,684
Georgian	3.6G	1.9G	171,950,621	91,569,739	Somali	61K	16K	1,202	472
German	308G	145G	44,878,908,446	21,529,164,172	South Azerbaijani	27M	19M	2,175,054	1,528,709
Goan Konkani	2.2M	1.8M	124,277	102,306	Spanish	278G	149G	47,545,122,279	25,928,290,729
Guarani	36K	24K	7,382	4,680	Sundanese	211K	141K	30,321	20,278
Gujarati	1.1G	722M	72,045,701	50,023,432	Swahili	13M	8.1M	2,211,927	1,376,963
Haitian	3.9K	3.3K	1,014	832	Swedish	44G	25G	7,155,994,312	4,106,120,608
Hebrew	20G	9.8G	2,067,753,528	1,032,018,056	Tagalog	573M	407M	98,949,299	70,121,601
Hindi	17G	8.9G	1,372,234,782	745,774,934	Tajik	379M	249M	31,758,142	21,029,893
Hungarian	40G	18G	5,163,936,345	2,339,127,555	Tamil	9.3G	5.1G	420,537,132	226,013,330
Icelandic	1.5G	846M	219,900,094	129,818,331	Tatar	670M	305M	51,034,893	23,825,695
Ido	147K	130K	25,702	22,773	Telugu	2.5G	1.6G	123,711,517	79,094,167
Iloko	874K	636K	142,942	105,564	Thai	36G	16G	951,743,087	368,965,202
Indonesian	30G	16G	4,574,692,265	2,394,957,629	Tibetan	187M	138M	1,483,589	936,556
Interlingua	662K	360K	180,231	100,019	Tosk Albanian	5.0M	2.8M	841,750	459,001
Interlingue	24K	1.6K	5,352	602	Turkish	60G	27G	7,577,388,700	3,365,734,289
Irish	88M	60M	14,483,593	10,017,303	Turkmen	11M	6.8M	1,113,869	752,326
Italian	137G	69G	22,248,707,341	11,250,012,896	Tuvinian	12K	7.9K	759	540
Japanese	216G	106G	4,962,979,182	1,123,067,063	Uighur	122M	83M	8,657,141	5,852,225
Javanese	659K	583K	104,896	86,654	Ukrainian	53G	28G	4,204,381,276	2,252,380,351
Kalmyk	113K	112K	10,277	10,155	Upper Sorbian	4.2M	1.8M	545,351	236,867
Kannada	1.7G	1.1G	81,186,863	49,343,462	Urdu	2.7G	1.7G	331,817,982	218,030,228
Karachay-Balkar	2.6M	2.3M	185,436	166,496	Uzbek	21M	12M	2,450,256	1,381,644
Kazakh	2.7G	1.5G	191,126,469	108,388,743	Venetian	18K	17K	3,492	3,199
Kirghiz	600M	388M	44,194,823	28,982,620	Vietnamese	68G	32G	12,036,845,359	5,577,159,843
Komi	2.3M	1.2M	201,404	95,243	Volapk	2.0M	2.0M	321,121	318,568
Korean	24G	12G	2,368,765,142	1,120,375,149	Walloon	273K	203K	50,720	37,543
Kurdish	94M	60M	15,561,003	9,946,440	Waray	2.5M	2.2M	397,315	336,311
Lao	174M	114M	4,133,311	2,583,342	Welsh	213M	133M	37,422,441	23,574,673
Latin	26M	8.3M	4,122,201	1,328,038	Western Frisian	35M	26M	5,691,077	4,223,816
Latvian	4.0G	1.8G	520,761,977	236,428,905	Western Mari	1.2M	1.1M	93,338	87,780
Lezghian	3.3M	3.0M	247,646	224,871	Western Panjabi	12M	9.0M	1,426,986	1,111,112
Limburgan	29K	27K	4,730	4,283	Wu Chinese	109K	32K	11,189	4,333
Lithuanian	8.8G	3.9G	1,159,661,742	516,183,525	Yakut	42M	26M	2,547,623	1,789,174
Lojban	736K	678K	154,330	141,973	Yiddish	141M	84M	13,834,320	8,212,970
Lombard	443K	433K	75,229	73,665	Yoruba	55K	27K	8,906	3,518
Low German	18M	13M	2,906,347	2,146,417	Yue Chinese	3.7K	2.2K	186	128
Total	6.3T	3.2T	844,315,434,723	425,651,344,234					

Table 2: Size of the OSCAR corpus by language measured in bytes and number of words. Standard UNIX human-readable notation is used for the size in byte. We define "words" as spaced separated tokens, which gives a good estimate of the size of each corpus for languages using Latin or Cyrillic alphabets, but might give a misleading size for other languages such as Chinese or Japanese.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. *CoRR*, abs/1903.07785.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. *Technical report, Technical Report. Linguistic Data Consortium*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

Deduplication in Large Web Corpora

Vladimír Benko
Slovak Academy of Sciences, Ľ. Štúr Institute of Linguistics
vladimir.benko@juls.savba.sk

Abstract

Our paper tries to find answers to some questions related to deduplication process in large-scale web-crawled corpora. An experiment based on eight corpora from the Aranea family is introduced, and first results are presented.

1 Introduction

During past years, detection of duplicate data has been subject of increased research activity, motivated by efforts to save disk space in large-scale cloud-based storage systems (Mao et al., 2014), or to decrease size of index structures for web-based information system, such as search engines (Broder and Nelson, 1996; Zelenkov and Segalovich, 2007). In both cases, preference was given to algorithms capable of detecting duplicate data dynamically, i.e., such that evaluate each new document "as soon as it has arrived" (Waraporn et al., 2014).

In the context of corpus linguistics, the problem of duplicate data emerged relatively recently, mostly with the advent of the "Web as Corpus" research paradigm resulting in much larger corpora containing dramatically more duplicities. Due to the characteristics of a typical corpus processing pipeline, the detection of duplicates needs not to be performed for each document or text segment "on the fly", but rather the respective processing can be performed over the whole corpus (Pomikálek, 2011; Benko, 2013).

In both cases, it is obvious that detection of 100% duplicates is a relatively simple task, both from the theoretical and implementation perspective (Broder 1993), and the challenging part is the detection of near-duplicates (Pomikálek, op. cit.).

2 The Problem

Our paper will introduce a series of on-going experiments related to deduplication in large web-based corpora, in the framework of which we want to find answers to questions including (yet not limited to) as follows:

- How does the size of corpus influence the ratio of duplicate text segments of different level (documents, paragraphs and sentences).
- What are the optimal parameters of deduplication performed by the *Onion*[1] utility.
- What is the optimal method/metric for assessment the "quality" of deduplication.
- What is the nature of data that has been removed.

Our work is motivated mostly by the fact that we were able to find only very few papers devoted to these questions. In the framework of his PhD research, author of the *Onion* program based his evaluation of the deduplication process on counting the "surviving duplicate n-grams", and he worked with relatively small corpora only. The corpus used in Benko (2013) was larger, but still by at least one order of magnitude smaller than a typical web corpus. Moreover, that experiment had been performed on a traditional corpus with arguably different structure of duplicate phenomena in comparison with those in web corpora.

3 Deduplicating Aranea

For the first stage of our experiment, we decided to use data from our Aranea family of web corpora (Benko, 2014; Benko and Zakharov, 2016) that are not only sufficiently big but are also available in various source and intermediate formats suitable for the envisaged experiments.

As the deduplication of large corpora requires great amounts of computing resources (both RAM and processing time), corpus creators usually tend to optimize the process by opting for single pass and deduplicating on one type of text segment only, typically on paragraphs (Kilgarriff, 2014). In our case, however, we decided to perform the whole procedure in a progressive

[1] http://corpus.tools/wiki/Onion

manner, i.e., on the document, paragraph and sentence level, respectively. The advantage of such an approach is that the resulting corpora are available in several formats suitable for different types of use.

3.1 The Onion Pipeline

Onion is a mature, stable and extremely efficient tool optimized to detect and remove duplicate content for large-scale textual data files used in building language corpora. The way how it works is beyond the scope of this paper, and is described both in the already mentioned Pomikálek's dissertation, as well as in our previous work (Benko, 2013).

The program can basically work in two modes: by the default, the duplicates detected are simply deleted. Alternatively, duplicate text segments are only marked and the further decision what to do with them is left to an external utility – this was the functionality we used in the framework of our experiment.

3.2 "Onioning" the Paragraphs

As the input for our first experiment we used data of eight Aranea corpora, with four of them representing the "large" languages (English, French, German, and Russian), and the other four the "small" languages (Czech, Slovak, Swedish, and Latvian). Data of all these corpora had already been subject to standard pre-processing, such as filtration, tokenization, segmentation on sentences, and also document-level deduplication.

The standard Onion pipeline has been modified to produce continuous logging of the results (tokens in duplicate vs. non-duplicate text segments) after a user-settable threshold is reached (100 M by default). The deduplication was performed on 5-grams with a threshold of 0.9 and smoothing switched off[2], i.e., a text segment was considered duplicate if it contained over 90% n-grams already encountered in the previous text.

The results of paragraph-level deduplication are shown in Figure 1.

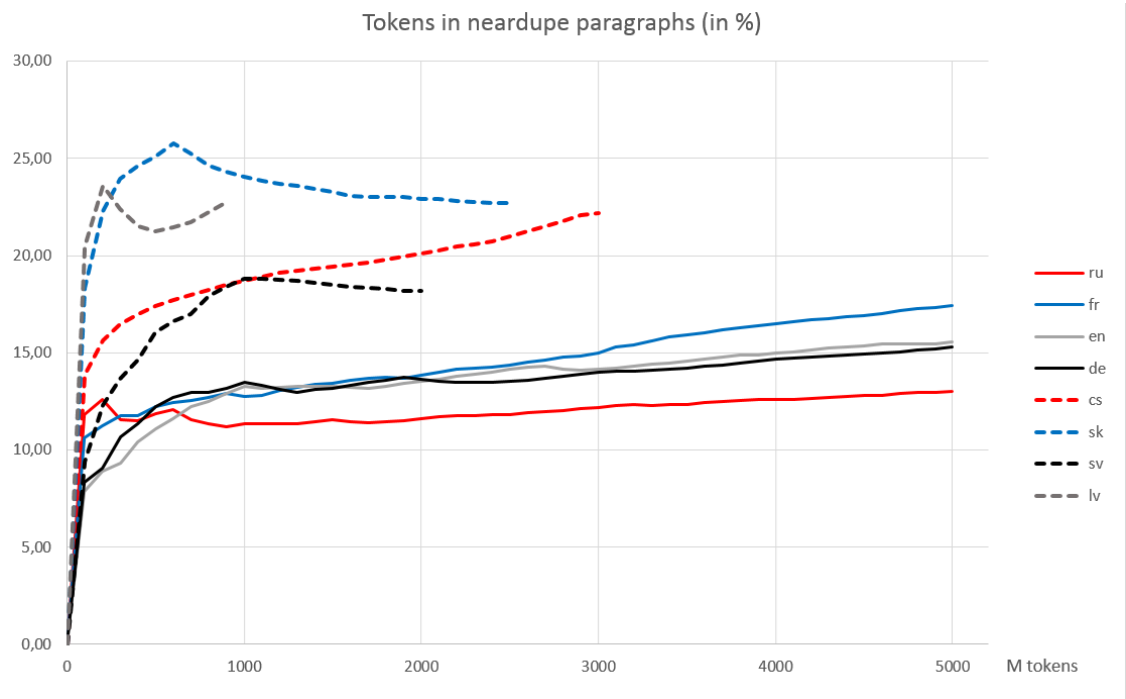


Figure 1: Deduplication on the paragraph level.

The result is somewhat surprising: it can be seen that the respective curves look very similar for the "large" languages, and after reaching the saturation, only small increase is observed. Although more data was available for these languages, we decided to cut the graph at the 5,000 Megatoken threshold to make the curves for "small" languages with less data more apparent.

The shape of curves for small languages is somewhat disparate, but we can observe that the ratios of duplicates are almost twice larger in comparison with "large" languages.

3.3 Deduping Sentences

Sentence-level deduplication is typically performed only in corpora that are to be analyzed by "reading", such as those used for lexicographic purposes. Duplicate sentences tend to negatively influence frequencies of

[2] In the smoothing mode, Onion also removes short non-duplicate segments between two duplicate ones.

lexical units and collocations, and impose additional burden for lexicographers compiling dictionary entries.

Lexicographers, however, belong to the "heaviest" users of our corpora (especially those containing the Slovak and Czech data), and sentence-level deduplication is therefore standard component of our processing pipeline.

The Figure 2. shows the result the process applied to the same eight corpora.

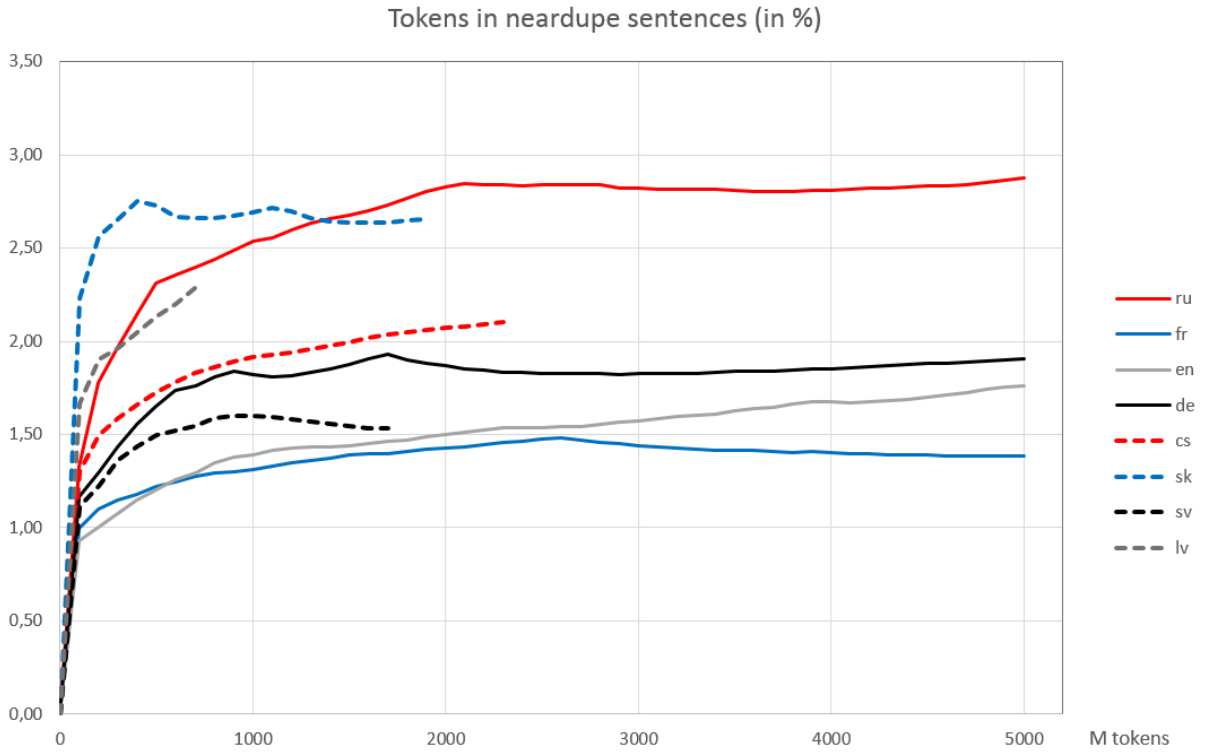


Figure 2: Deduplication on the sentence level.

Two phenomena can be observed in the figure. Firstly, the percentage of removed tokens is – not surprisingly – much smaller than in the paragraph-level deduplication. And secondly, the respective curves are much more similar, even for the "small" languages.

It might be quite interesting to observe that would happen if only sentence-level deduplication were performed – we'll probably make a new experiment targeted at this issue in the future.

4 Why Languages Differ

There are more ways how to examine the reasons of different "deduplication behavior" among languages involved of our experiment. Based on a suggestion of the anonymous reviewer of our paper, we decided to have a look at the number of Internet domains in the resulting corpora that could be used as a measure of data variety – the more different domains, the grater probability of differences in data.

To make the evaluations as simple as possible, we did not perform any new round of deduplication, and made use of the data already available: we produced frequency lists of Internet domains for all the *Minus* (100 Megaword) and *Maius* (1 Gigaword) versions of all the corpora involved. The results are shown in Table 1.

Language	**Domains**		**Ratio**
	Minus (100 MW)	**Maius** (1 GW)	
Russian	118,982	387,040	3.25
Czech	88,604	246,181	2.78
German	72,411	134,944	1.86
English	62,031	158,871	2.56
French	61,418	192,664	3.14
Slovak	49,738	126,024	2.53
Swedish	33,481	105,217	3.14
Latvian[3]	8,512	11,944	1.40

Table 1: Internet domains

[3] Only the *Parvus* class of corpus (530 MW) was available for Latvian.

The results are interesting but really need deeper analysis to be able to interpret the differences among the respective languages. It must be noted that several factors might have influenced the actual numbers – one of them being the number of crawling sessions that was varying from one or two for some languages to several dozens for the "featured" languages (Slovak, Czech and Russian).

5 What Data Has Been Removed

Our deduplication pipeline does not simply remove the duplicate content but rather splits the original file into two parts, i.e. retains the removed segments for possible further analysis. Due to the huge sizes of the respective files, this task is far from being easy. Here we show just a simple first step: finding the most frequent duplicate paragraphs and sentences.

As the Onion-based deduplication is performed on tokenized and tagged data, this procedure involves a reverse process, i.e., removing the annotation (lemma, tag and possible other attributes), "untokenizing" (converting vertical data to original one-paragraph-per-line format) and performing the respective frequency lists by means of standard *sort* and *uniq* utilities. The beginning of the resulting paragraph list is shown in Table 2.

Rank	Freq	Paragraph text
1	58,943	<s>Your email address will not be published.</s><s>Required fields are marked *</s>
2	55,739	<s>We've sent an email with instructions to create a new password.</s><s>Your existing password has not been changed.</s>
3	52,223	<s>It looks like you're already registered</s>
4	44,816	<s>Save changes Preview Cancel</s>
5	26,758	<s>Your password has been changed</s>
6	26,757	<s>Password has been successfully updated.</s>
7	26,619	<s>Conference Presentation Video</s>
8	26,149	<s>Email address is required.</s>
9	26,113	<s>Enter your email and we'll send you a link to reset your password.</s>
10	26,112	<s>You're almost there.We've just sent a confirmation email to .</s><s>Check it out to confirm your registration.</s>
11	26,112	<s>We have sent a confirmation email to .</s><s>Please check your email and click on the link to activate your account.</s>
12	26,112	<s>We are unable to process your request at this time.</s><s>Please try again later.</s>
13	26,112	<s>Thank you for registering</s>
14	26,112	<s>Please fill in the remaining fields below to complete your registration</s>
15	26,112	<s>It looks like you're already registered.</s>
16	26,112	<s>is already registered with .</s><s>You will be able to use the same account on .</s><s>Alternatively, you can create a new account with another email address.</s>
17	26,112	<s>Congratulations, you've just sealed the deal!</s><s>Sign in to your profile now to get started.</s>
18	26,112	<s>By registering you are agreeing to the Terms and Conditions of the website.</s>
10	26,111	<s>We didn't recognise that password reset code.</s><s>Enter your email address to get a new one.</s>
20	26,111	<s>We are unable to send your welcome email at this time.</s><s>Please try again later by clicking the resend welcome email link from your profile page.</s>

Table 2: Most frequent duplicate paragraphs (English)

As it can be seen, the most frequent dupes are surprisingly quite long and apparently come from very similar texts – at least their frequencies suggest so.

The Table 3 shows similar list resulting from the sentence-level deduplication. The situation here is different – the most frequent "sentences" are in fact short text fragments, and some of them even raise questions about appropriateness of the sentence segmentation policy.

Rank	Freq	Sentence text
1	532,867	<s>1.</s>
2	477,841	<s>2.</s>
3	407,229	<s>3.</s>
4	315,925	<s>4.</s>
5	247,789	<s>5.</s>
6	181,323	<s>6.</s>
7	145,202	<s>7.</s>
8	117,650	<s>8.</s>

9	98,438	<s>9.</s>
10	92,226	<s>Why?</s>
11	91,129	<s>.</s>
12	85,738	<s>10.</s>
13	72,879	<s>Read more</s>
14	60,538	<s>Read More</s>
15	60,327	<s>Yes.</s>
16	59,769	<s>More</s>
17	58,953	<s>11.</s>
18	54,932	<s>Abstract</s>
10	52,645	<s>a.</s>
20	51,510	<s>12.</s>
21	49,238	<s>1</s>
22	46,641	<s>–</s>
23	46,616	<s>b.</s>
24	43,727	<s>13.</s>
25	42,615	<s>You are here</s>
26	42,460	<s>3</s>
27	40,405	<s>2</s>
28	39,024	<s>14.</s>
29	37,228	<s>Description</s>
30	37,005	<s>Comments</s>
32	36,643	<s>MR.</s>
32	36,521	<s>Pages</s>

Table 3: Most frequent duplicate sentences (English)

The optimal strategy for analyzing the files containing duplicate data is yet to be developed and may also depend on the expected use of the resulting corpus. For lexicographic use, for example, one of the promising options may be looking for lexical units present in duplicate data, yet missing in the deduplicated corpus, with the amount of them being used as a measure of the "quality" of deduplication.

6 Conclusion and Further Work

It is probably too early to make any final conclusions before this experiment is performed with more data and more parameters for the *Onion* program, perhaps also with finer logging thresholds to see the shape of the curve before the saturation.

What can be, however, said after this first stage of our experiment is that the amount of data removed during deduplication depends on many factors associated not only with the respective language itself, but also with the size of "searchable web" for the respective language.

Acknowledgments

This work has been, in part, funded by the Slovak KEGA and VEGA Grant Agencies, Project No. K-16-022-00, and 2/0017/17, respectively.

References

Vladimir Benko. 2013. *Data Deduplication in Slovak Corpora.* In Slovko 2013: Natural Language Processing, Corpus Linguistics, E-learning. RAM-Verlag: Lüdenscheid, 2013, pp. 27-39.

Vladimir Benko. 2014. *Aranea: Yet Another Family of (Comparable) Web Corpora.* In Petr Sojka, Aleš Horák, Ivan Kopeček and Karel Pala (Eds.): Text, Speech and Dialogue. 17th International Conference, TSD 2014, Brno, Czech Republic, September 8-12, 2014. Proceedings. LNCS 8655. Springer International Publishing Switzerland, 2014. pp. 257-264. ISBN: 978-3-319-10815-5 (Print), 978-3-319-10816-2 (Online).

Vladimír Benko, and Victor P. Zakharov. 2016. *Very Large Russian Corpora: New Opportunities and New Challenges.* In Kompjuternaja lingvistika i intellektual'nye tekhnologii: Po materialam mezhdunarodnoy konferentsii «Dialog» (2016), vypusk 15 (22). Moskva: Rossijskiy gosudarstvennyy gumanitarnyy universitet, 2016, pp. 79–93.

Andrei Z. Broder. 1993. *Some applications of Rabin's fingerprinting method.* In: Sequences II: Methods in Communications, Security, and Computer Science. Springer-Verlag. http://xmail.eye-catcher.com/rabin_apps.pdf.

Adam Kilgarriff. 2014. *Personal communication.*

Bo Mao, Hong Jiang, Suzhen Wu, Yinjin Fu, Lei Tian. 2014. *Read-Performance Optimization for Deduplication-Based Storage Systems in the Cloud.* In ACM Transactions on Storage (TOS). Volume 10 Issue 2, March 2014. https://doi.org/10.1145/2512348

Jan Pomikálek. 2011. *Removing Boilerplate and Duplicate Content from Web Corpora.* Ph.D. thesis. Masaryk University in Brno, Faculty of Informatics. http://is.muni.cz/th/45523/fi_d/phdthesis.pdf.

Yuriy G. Zelenkov, and Ilya V. Segalovich. 2007. *Sravnitel'nyj analiz metodov opredeleniya nechetkikh dublikatov dlya Web-dokumentov* (*Comparative analysis of near-duplicate detection methods of Web documents*). In Trudy 9-oj Vserossijskoj nauchnoj konferencii «Elektronnye biblioteki: perspektvinye metody i tekhnologii» RDCL 2007. Perelsavl'-Zalesskij: «Universitet goroda Pereslavlya», 2007. pp. 166–174.

Waraporn Leesakul, Paul Townend, and Jie Xu. 2014. *Dynamic Data Deduplication in Cloud Storage.* In IEEE 8th International Symposium on Service Oriented System Engineering. 7-11 April 2014, Oxford, UK. https://doi.org/10.1109/SOSE.2014.46

The best of both worlds: Multi-billion word "dynamic" corpora

Mark Davies
Department of Linguistics
Brigham Young University
mark_davies@byu.edu

Abstract

Nearly all of the very large corpora of English are "static", which allows a wide range of one-time, pre-processed data, such as collocates. The challenge comes with large "dynamic" corpora, which are updated regularly, and where pre-processing is much more difficult. This paper provides an overview of the NOW corpus (News on the Web), which is currently 8.2 billion words in size, and which grows by about 170 million words each month. We discuss the architecture of NOW, and provide many examples that show how data from NOW can (uniquely) be extracted to look at a wide range of ongoing changes in English.

1 Corpus architecture

Multi-billion word corpora have become commonplace in the last 5-10 years. For example, there are several different 10-20 billion word corpora from Sketch Engine (Kilgarrif et al 2014; www.sketchengine.eu), Corpora from the Web (Schäfer 2015; corporafromtheweb.org), and English-Corpora.org (formerly the BYU Corpora).

Most of these corpora, however, are "static" corpora. The corpus texts are collected and annotated, and they are then indexed and pre-processed in other ways, which makes text retrieval very fast even on very large corpora. For example, the 14 billion word iWeb corpus (https://www.english-corpora.org/iweb), users can search by word form, lemma, part of speech, synonyms, user-defined wordlists, and more. A search for a complex string like *VERB _a =EXPENSIVE @CLOTHES* (verb + article + any form of any synonym of *expensive* + any form of any word in the user-defined *clothes* wordlist) will take just 2-3 seconds.

iWeb and all of the corpora from English-Corpora.org are based on highly-optimized relational databases, which yields corpora that are typically 5-10 times as fast as other large corpora (see www.english-corpora.org/speed.asp). The underlying architecture is similar to "columnstore" databases. In a 14 billion word corpus, for example, there would be 14 billion rows, each with a structure like the following:

ID	textID	word9	word10	word11	word12	word13
536495784	199	143	122	1983	181	4096161
535599496	1497	16	6	1983	687	2
535389538	2098	2	20	1983	271	5
535969715	2199	5	85	1983	1052	9
536189340	3999	85	122	1983	1201	1
535977462	5297	12	6	1983	634	2
535976705	5297	6	122	1983	634	2
535419837	5876	3342	36	1983	177	35
536545169	6094	1808	6	1983	1911	2

Figure 1: Corpus architecture

Each word / lemma / PoS combination is represented as an integer value, which is tied to an entry in the lexicon (and which is in a separate database). In Figure 1, for example, the integer value [1983] represents [best / best / jjt]. There is a clustered index on this "middle" column ([word11] in Figure 1), which means that all of the tokens of any word (*best* in this case) are stored *physically* adjacent to each other on the SSD, which increases access speed a great deal.

As it carries out the search, iWeb (or any of the corpora from English-Corpora.org) parses the search string to find the lowest-frequency, "weakest" part of the string. For example, in the search string *the best NOUN*, the word *best* occurs less than either *the* or all NOUNs. The search focuses first on the lemma *best*, and only when it finds those rows (all of the rows containing the value 1983 in column [word11]) does it narrow this to rows where the preceding column ([word10] in Figure 1) is the value for *the* and the following column ([word12] in Figure 1) is an integer value tied to a noun in the lexicon. (Note that in Figure 1 (for reasons of space), only the two columns to the left and to the right of the "node" column are shown, but – depending on the corpus – there are 5-10 columns each to the left and to the right).

Davies (2019) explains the underlying architecture in more detail, and provides a number

of examples that show that the corpora with this architecture are typically 5-10 times as fast as the architecture of other very large corpora. Crucially, this is because these other corpora typically parse the search string left to right (e.g. with the word *the* first in the string *the best NOUN*), whereas we focus first on the "weakest link" in the search string.

Our approach also takes full advantage of relational database architecture, such as JOINs across any number of highly-optimized tables. For example, in the example of *VERB _a =EXPENSIVE @CLOTHES* shown above (verb + article + any form of any synonym of *expensive* + any form of any word in the user-defined *clothes* wordlist), the search will use lemma and part of speech information from the main [lexicon] table, as well as a separate [synonyms] table containing entries for more than 65,000 words, and another table containing user-defined lists such as clothing, emotions, or a particular class of verbs. Additional tables could contain pronunciation information or additional semantic information, and the search speed will not decrease much (if at all) no matter how many tables are involved.

Finally, there is a [sources] table that can contain any number of columns related to each of the texts in the corpus, and these are JOINed to the main corpus table (e.g. Figure 1) via the [textID] value. This allows users to quickly and easily create "virtual corpora" using any of the metadata from the [sources] table, such as author, date, website, or genre.

When the corpus sees that all of the "slots" in a search are very frequent, it defaults to using pre-processed n-grams, which are even faster than the previous approach. For example, a very high frequency search like "NOUN NOUN" takes less than two seconds, because it is only searching 10 or 100 million rows of data in the n-grams databases. (The downside of the n-gram tables is that they refer to the entire corpus, and not just particular sections, just as certain genres or texts.)

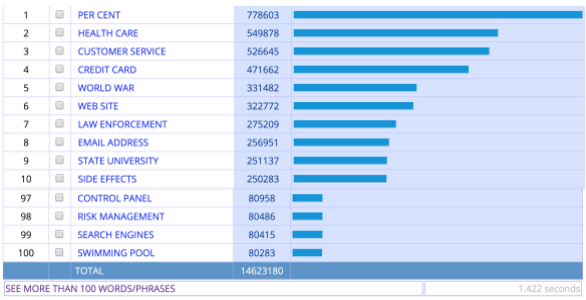


Figure 2: iWeb high frequency: *NOUN + NOUN*

Finally, as with the Sketch Engine corpora, other data such as collocates are pre-processed in iWeb, which means they can be retrieved in just a second or two.

Figure 3: iWeb collocates for *bread*

Pre-processing also allows for very fast retrieval (1-2 seconds for results from the 14 billion word corpus) for word clusters, related topics (words that frequently co-occur anywhere on the 22 million web pages), websites that use the word the most (which can be used to quickly and easily create "Virtual Corpora" on almost any topic), and sample concordance lines (see Davies 2019).

2 Creating the dynamic NOW corpus

As we will discuss in Section 4. the challenge comes, however, when we create a corpus that is "dynamic. (We define "dynamic" as corpora in which texts are continually added, rather than corpora in which texts are both added and deleted – although our architecture would have the same advantages in this case as well.)

An example of a dynamic corpus is the NOW Corpus ("News on the Web"; www.english-corpora.org/now), which is – as far as we are aware – the only corpus larger than a billion words, and which is growing on a regular basis (at least every month). The NOW corpus debuted at 3.6 billion words in May 2016 (with texts going back to 2010) and is now (early July 2019) about 8.2 billion words in size. Every month 150-170 million words are added to the corpus, or about 1.5 billion words each year. Note that similar corpora for Spanish and Portuguese are also available (corpusdelespanol.org/now: 6.0 billion words in 21 Spanish-speaking countries since 2012, and corpusdoportugues.org/now: 1.3 billion words in 4 Portuguese-speaking countries since 2012), but the English NOW corpus will be the focus of this paper.

To create the NOW corpus, every hour five different machines search Google News to retrieve newly-listed newspaper and magazine articles, for 20 different English-speaking countries (the same 20 countries as GloWbE; see Davies 2013). For example, Figure 4 shows just

two sample entries from Google News from 3 July 2019, and on average we gather the URLs for about 20,000 such articles each day.

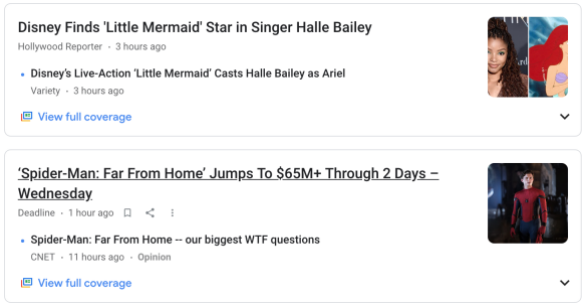


Figure 4: Sample Google News entries

The metadata for each of the 20,000 articles (URL, title, source, Google snippet) that appear each day are stored in a relational database. For example, the following is a small selection of the links from Google News from the US and Canada for the last hour on April 24, 2019, as the initial version of this paper was being written:

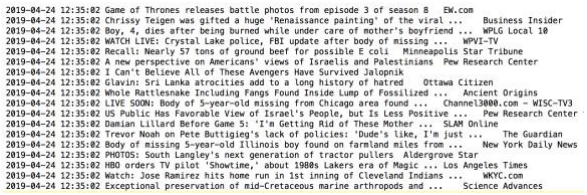


Figure 5: NOW sample list of articles

At the end of the month, we download the 250,000-300,000 articles using a custom program written in the Go language, which downloads all of the 250,000_ texts in about 30-40 minutes. We then use JusText (Pomikálak 2011; corpus.tools/wiki/Justext) to remove boilerplate material, and we tag the text with CLAWS 7 (for English; see Garside and Smith 1997), and a customized tagger based on Eckhard Bick's Palavras tagger for the Portuguese and Spanish corpora (Bick 1999). We then remove duplicate articles (always a problem in newspaper-based corpora) by looking for duplicate 11-grams across texts. For example, if a text has 68 11-grams starting with the word *the*, and 39 of these 11-grams are also found in any of the other 250,000+ texts from that month, then the text is tagged as a probable duplicate and it is removed from the corpus. (This process takes only 2-3 minutes for the 150-170 million words, because of the relational database architecture underlying the corpus).

Once we have done all of these steps, the new texts are then added to the existing corpus. As the Figure 6 shows (for Nov 2018 – June 2019), this results in about 150-175 million additional words of data each month:

	# WEBSITES	# TEXTS	# WORDS	TOTAL = 8,157,007,165 WORDS
2019 June	7,561	323,438	171,418,865	
2019 May	7,298	313,771	175,811,655	
2019 Apr	7,281	311,756	167,682,945	
2019 Mar	7,125	307,939	176,732,360	
2019 Feb	6,456	309,978	164,462,989	
2019 Jan	6,698	330,215	173,665,746	
2018 Dec	6,416	272,799	147,540,743	
2018 Nov	7,234	298,425	160,186,292	

Figure 6: NOW size by month (last 8 months)

Note that NOW contains just those articles that Google News links to, which are primarily newspaper and magazine sites. But there is an incredible variety in these sites – they are not just "staid" broadsheet newspapers. They include magazine and newspaper articles dealing not only with current events, but also technology, entertainment, and a wide variety of topics (as is evidenced by the 7,000+ "news" sites in a given month, as shown in Figure 6).

Evidence for the often informal nature of the texts comes from an investigation of the lexical creativity in the corpus. For example, there are more than 540 different *–alypse* words that are formed by analogy to the word *apocalypse*, such as *snarkpocalypse*, *snowpocalypse, chocopalypse, crapocalypse, kittiepocalypse, redditpocalypse, zombiepocalypse, and biebopalypse.* Likewise, there are more than 4,400 –fest words, including such innovative words as *gloomfest, testosterone-fest, brixfest, weep-fest, rant-fest, glumfest, oktemberfest, foul-fest,* and *raunchfest* (all of which occur at least five times in the corpus).

3 Examples from the NOW corpus

The advantage of a dynamic "monitor" corpus like NOW is that we are able to see what is going on with the language at the current time – not just 2 or 5 or 10 years ago.

At the most basic level, users can search for the frequency of a given word or phrase since 2010. For example, the following are just a few of the new words and phrases since 2010: *Brexit, trigger warning, catfishing, nomophobia, FOMO, birther, selfie stick, data lake, digital native, ransomware.* Some other cases of increase since 2010 include: (NOUN) *refugee, ransomware* (ADJ) *transgender*, self-driving, on-demand, streaming, far-right* (VERB) *overreach, eventuate, intensify, text, retweet* (ADV) *effectively, programmatically*. Words showing a decrease in use during this time include: (NOUN)

waitress, disc, fax (ADJ) *neat, old-fashioned, eco-friendly, eco-conscious, loopy, preppy, sullen, scanty* (VERB) *cream, clunk, flunk, gripe, murmur, foreclose* (ADV) *honorably, contentedly, frightfully*.

For any of these words or phrases, the NOW corpus shows the frequency in six month blocks (and with even more granularity, as we will soon see). For example, Figure 7 shows the decreasing frequency of *waitress* (which is viewed by some as being sexist, because of the feminine *–ess* ending) almost year by year since 2010:

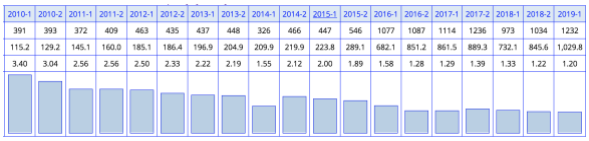


Figure 7: Frequency of *waitress*: every 6 months

The 497,000+ tokens of *Brexit* show that it increased suddenly in the first half of 2016, and that (after a bit of a pause in late 2017 and early 2018) it has increased again in early 2019, to its highest level yet:

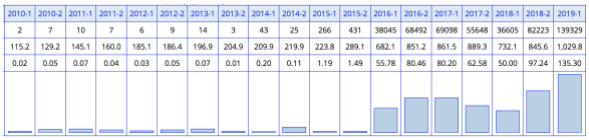
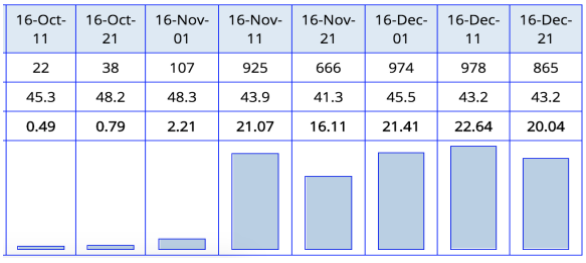


Figure 8: Frequency of *Brexit*: every 6 months

It is also possible to see the frequency of a word or phrase in 10-day increments. For example, the NOW corpus shows that the phrase *fake news* comes out of nowhere within a day or two of the 2016 US presidential elections (Nov 8, 2016):



16-Oct-11	16-Oct-21	16-Nov-01	16-Nov-11	16-Nov-21	16-Dec-01	16-Dec-11	16-Dec-21
22	38	107	925	666	974	978	865
45.3	48.2	48.3	43.9	41.3	45.5	43.2	43.2
0.49	0.79	2.21	21.07	16.11	21.41	22.64	20.04

Figure 9: Frequency of *fake news* by 10 day period

The NOW corpus can also be used to examine cultural shifts. For example, Google Trends (which measures the frequency of searches, but not the actual frequency of a word or phrase in texts), shows that people started searching for *fidget spinner* in April 2017, that it reached its peak in mid-May 2017, and that it largely disappeared by June/July 2017. The NOW corpus (Figure11; based on actual occurrences in texts) shows the same thing:

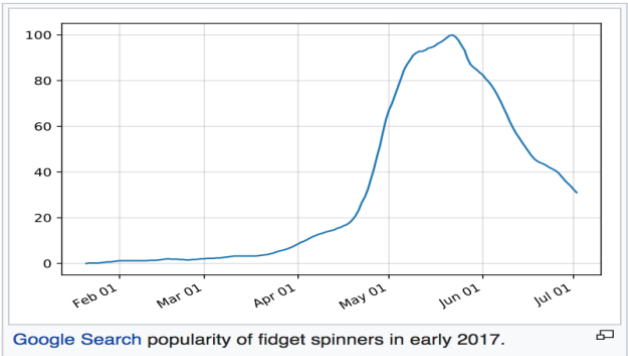


Figure 10: *fidget spinner* in Google Trends

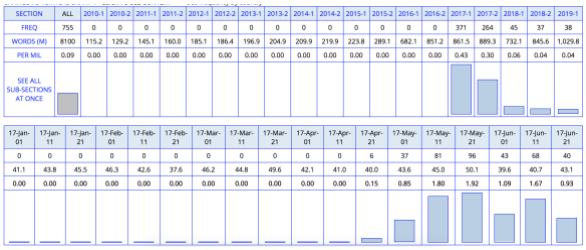


Figure 11: *fidget spinner* in NOW by 10 day period

3.1 The corpus architecture also allows users to quickly and easily compare the results in one section (e.g. a particular time period) to those of another section (or time period) (see Davies 2017, 2018 for many more examples). For example, the following chart shows words ending in **gate* (sometimes indicating "scandal") that are more frequent in 2017-2019 (top; e.g. *Panamagate, dieselgate, deflategate*) compared to 2010-2013 (bottom; e.g. *hackgate, cablegate, climategate*):

SEC 2 (2017-1, 2017-2, 2018-1, 201...): 4,358,255,771 WORDS

	WORD/PHRASE	TOKENS 2	TOKENS 1	PM 2	PM 1	RATIO
1	LANGATE	323	1	0.1	0.0	68.3
2	PANAMAGATE	2298	0	0.5	0.0	52.7
3	NIXON/WATERGATE	167	1	0.0	0.0	35.3
4	DIESELGATE	1342	0	0.3	0.0	30.8
5	GAMERGATE	351	4	0.1	0.0	18.5
6	PIZZAGATE	526	8	0.1	0.0	13.9
7	DEFLATEGATE	571	0	0.1	0.0	13.1

SEC 1 (2010-1, 2010-2, 2011-1, 201...): 920,939,433 WORDS

	WORD/PHRASE	TOKENS 1	TOKENS 2	PM 1	PM 2	RATIO
1	TRI-GATE	99	1	0.1	0.0	468.5
2	SUMMERGATE	17	1	0.0	0.0	80.5
3	WEINERGATE	31	2	0.0	0.0	73.4
4	HACKGATE	30	2	0.0	0.0	71.0
5	CLIMATEGATE	431	47	0.5	0.0	43.4
6	ENERGATE	18	2	0.0	0.0	42.6
7	HYGATE	30	4	0.0	0.0	35.5

Figure 12: Comparison of **gate* words 2017-2019 (top) vs 2010-2012 (bottom)

And of course researchers can compare new phrases as well (rather than just words). For example, the following are all new phrases with smart NOUN that are at least 20 times as frequent

in 2017-2019 as they were in 2010-2013 (if they occur back then at all): *smart speaker, smart pole, smart airport, smart workplace, smart condom, smart coating, smart gas, smart doorbell, smart shower, smart park, smart waste,* and *smart fence*.

3.2 In addition to looking at changes in lexis and phraseology, researchers can also use NOW to look at very recent changes in syntax. The impression has often been that syntax changes so slowly that a corpus with just a ten year time span (as with NOW; 2010-2019) wouldn't show much change during this short period. But cases of syntactic change during just the last ten years are not hard to find

For example, the frequency of the perfect progressive (HAVE+been+VERB-ing: *has been working*) has increased about 10% during the last ten years, from less than 260 tokens per million words in 2010-2011, to 280-290 tokens per million words in 2017-2019.

Likewise, there have been changes in verbal subcategorization during just the last few years. For example, Figure 13 shows an increase in the "bare infinitive" with *help* (e.g. *they helped me -- clean the room*) compared to the "to infinitive" (*they helped me **to** clean the room*) since 2010. (The figure shows the percentage of all tokens that are the bare infinitive. For more on the construction, which has been a favorite of corpus linguistics, see Kjellmer 1985, Mair 2002, Rohdenburg 2009, and Callies 2015.)

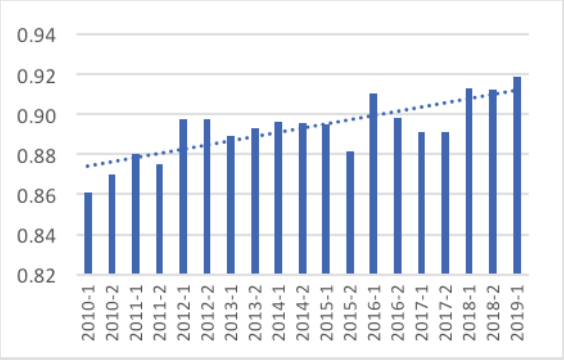


Figure 13: % HELP PRON -- VERB

Finally, it is possible to see change in just a given variety (or group of varieties) of English, such as British, American, or Singaporean English. For example, Figure 14 shows the increase in *gotten* as a past participle (e.g. *I've gotten over the guilt*) compared to the more common *got* (*I've got over the guilt*) in British English.

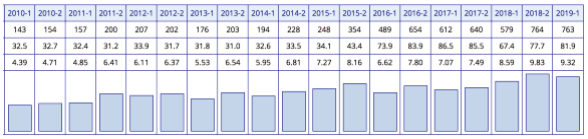


Figure 14: (HAVE+) *gotten* in British English

Whereas the normalized frequency was less than five tokens per million words in 2010-2011, it is nearly twice that (8.6 to 9.8 tokens per million words) in 2018-2019. Because we can focus on both different time periods and different varieties in NOW, we can use the corpus to see how linguistic changes spread from one dialect to another over time.

In summary, NOW allows us to look at ongoing changes in English in ways that are not possible with any other corpus. This is due to two features that NOW has, which are not found together in any other corpus – its very large size and the fact that it has been updated on a regular basis (every month), up to the current time.

4 Problems and challenges

In spite of the possibilities with a continually updated corpus like NOW, there are also some challenges – compared to "static" corpora like iWeb.

First, as was explained in Section 2, the SQL Server database relies heavily on "clustered" indexes for search speed. This means that data is physically stored on the SSD – one row next to another – according to whatever column we choose. Therefore, when new data is added to the corpus (for example, 170-180 million words each month for NOW), the new rows of data need to be placed (on the SSD) adjacent to the existing rows. For example, all of the rows for the word *market* need to be physically placed between *market* and the next word (such as *marketable*). If the "fill factor" is not set high enough, millions of rows of data will need to be moved on the SSD to make room for the new rows of data. This can be very slow, even for SSDs.

Second, in iWeb we could create n-gram databases to handle very high frequency searches, like "VERB the NOUN" or "NOUN NOUN". With the NOW corpus, we would need to rebuild these every time the corpus is updated, such as every month. Because the corpus is now so large (more than 8 billion words), this would be computationally quite expensive to do each month. As a result, we do not use n-grams for NOW, which means that some very high frequency search strings (e.g. NOUN NOUN) are disallowed.

Third, there is other data that is pre-processed in iWeb that would be expensive to pre-process every month in NOW, such as collocates. The only reason that collocates are even doable in iWeb or the Sketch Engine corpora is because they *are* pre-processed. But the collocates would need to be pre-processed again for all 60,000 lemmas whenever new data is added to the corpus, and that can take a full day or two. And unless the collocates are re-generated each month, the collocates data will gradually become more and more outdated until they are updated again.

One might claim that in principle other architectures that are designed for "static" corpora *should* be able to use preprocessing strategies for incrementally updated values (such as ngram indices or term frequencies). But we are not aware of any other very large corpora that *actually employ* such an approach, for corpora that are updated every day or even every month. And while term frequencies can be easily updated, other data such as collocates and n-grams will take a significant amount of time, to say nothing of the basic "clustered" data, as explained above.

5 Conclusion

In summary, the NOW corpus provides at least two important advantages. First, it is very large – currently more than 8 billion words in size. Second, unlike most other large corpora, it is continually updated – by about 150-170 million words each month, or 1.5 billion words each year. The combination of these two features allows it to model ongoing linguistic change in English in ways that are not possible with any other corpus.

Due to its relational database architecture (which uses an architecture similar to sharding in columnstore databases, including clustered indexes), most searches (words, substrings, phrase, and even grammatical constructions; cf. "HELP PRON (to) VERB" shown above) are only 4-5% slower in an 8 billion word corpus (the current size of NOW) than in a 3-4 billion word corpus (the size of NOW in 2015).

But some searches (such as very high frequency strings like NOUN NOUN, which are based on n-grams), or queries that use pre-processed data (such as collocates) can still present a challenge in these dynamic corpora.

References

Bick, Eckhard. 1999. *The parsing system Palavras*, Aarhus: Aarhus Univ. Press.

Callies, Marcus. 2013. Bare infinitival complements in Present-Day English. In *The Verb Phrase in English: Investigating Recent Language Change with Corpora*. Ed. Bas Aarts, et al. Cambridge: CUP, 239-255.

Davies, Mark and Jong-Bok Kim. 2019. The advantages and challenges of 'big data': Insights from the 14 billion word iWeb corpus. *Linguistic Research* 36: 1-34.

Davies, Mark. 2018. Corpus-based studies of lexical and semantic variation: The importance of both corpus size and corpus design. In *From data to evidence in English language research*. Ed. Carla Suhr, et al. Leiden: Brill. 34-55.

Davies, Mark. 2017. Using Large Online Corpora to Examine Lexical, Semantic, and Cultural Variation in Different Dialects and Time Periods. In *Corpus-Based Sociolinguistics*. Ed. Eric Friginal et al. London: Routledge. 19-82.

Davies, Mark and Robert Fuchs. 2015 Expanding Horizons in the Study of World Englishes with the 1.9 Billion Word Global Web-Based English Corpus (GloWbE). *English World-Wide* 36: 1-28.

Garside, R., and Smith, N. 1997. A hybrid grammatical tagger: CLAWS4/ *Corpus Annotation: Linguistic Information from Computer Text Corpora*. Ed. Roger Garside, et al. Longman, London, pp. 102-121

Kilgarriff, Adam, Pavel Rychlý, Pavel Smrž, David Tugwell. Itri-04-08 the sketch engine. *Information Technology*, 2004.

Kjellmer, Göran. 1985. Help to/help – revisited. *English Studies* 66: 156-61.

Mair, Christian. 2002. Three Changing Patterns of Verb Complementation in Late Modern English: A Real-Time Study Based on Matching Text Corpora/ *English Language and Linguistics* 6: 105-131.

Pomikálek, Jan. 2011. Removing Boilerplate and Duplicate Content from Web Corpora. Ph.D. thesis, Univ. Masaryk.

Rohdenburg, Gunter. 2009. Grammatical Divergence between British and American English in the Nineteenth and Early Twentieth Centuries. *Linguistic Insights - Studies in Language and Communication* 77:301-329.

Schäfer, Roland. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3)*, Ed. Piotr Banski, et al.

The Vast and the Focused: On the need for thematic web and blog corpora

Adrien Barbaresi
Berlin-Brandenburg Academy of Sciences
Jägerstraße 22/23
D-10117 Berlin
barbaresi@bbaw.de

Abstract

As the Web ought to be considered as a series of sources rather than as a source in itself, a problem facing corpus construction resides in meta-information and categorization. In addition, we need focused data to shed light on particular subfields of the digital public sphere. Blogs are relevant to that end, especially if the resulting web texts can be extracted along with metadata and made available in coherent and clearly describable collections.

1 Problem description

The Web brings an unparalleled and rapidly evolving diversity in terms of speakers and settings. As such it should be considered as a series of sources rather than as a source in itself. Science needs an agreed scheme for identifying and registering research data (Sampson, 2000), in that sense schemes and methods are needed to live up to the potential of these potential sources for corpus construction. "Offline corpora" accessible within or throughout institutions are now standard among the research community. The process notably involves "crawling, downloading, 'cleaning' and deduplicating the data, then linguistically annotating it and loading it into a corpus query tool." (Kilgarriff, 2007) It relies on the assumption that "the Web is a space in which resources are identified by Uniform Resource Identifiers (URIs)." (Berners-Lee et al., 2006) The Web is however changing faster than the researchers' ability to observe it (Hendler et al., 2008), and a constant problem faced by web resources resides in meta-information and categorization. Due to the "heteregeneous and somewhat intractable character of the Web" (Bergh and Zanchetta, 2008), the actual contents of a web corpus can only be listed with certainty once the corpus is complete. In addition, web corpora exemplify "problems of large corpora built in short time and with little resources." (Baroni and Ueyama, 2006)

In fact, corresponding to the potential lack of information concerning the metadata of the texts is a lack of information regarding the content, whose adequacy, focus and quality has to be assessed in a post hoc evaluation (Baroni et al., 2009). The ability to describe a corpus accurately significantly increases its interest for researchers in the humanities and beyond. This is neither a trivial task nor a secondary one, as some assume that "text category is the most important organizing principle of most modern corpora" (O'Keeffe and McCarthy, 2010). Renouf (2007) also claims that lack of metadata makes an exhaustive study impossible or at least undermines it. Categories such as audience, authorship and artifact (Warschauer and Grimes, 2007), or authorship, mode, audience, aim, domain, and the annotation of textual dimensions (Sharoff, 2018) target this issue in particular.

Besides, a major fault line exists for the linguistic community between general and specific corpora (Gries, 2009). Since web corpora mostly follow from the existing linguistic tradition, their purpose and their methodology can also be divided into two main categories (Barbaresi, 2015). On the one hand there are all-purpose, "one size fits all" corpora, often designed to be large and diverse. On the other, there are specific corpora with controlled text inclusions and possibly rich metadata, built with particular research goals in mind, such as online news corpora or variation-aware approaches which take production conditions into account. This distinction also overlaps with diverging uses for corpora, for example corpus-based studies observing already known phenomena, and more opportunistically-minded research settings where size and content diversity allow for better coverage and use of statistical indicators. The contrast between general-purpose

and specific corpora is not clear-cut as these categories are not impermeable: it is possible to find corpora that are in-between, or transferred from one to another due to later developments in corpus design.

2 From the vast to the focused

Seen from a practical perspective, the purpose of focused web corpora is to complement existing collections, as they allow for better coverage of specific written text types and genres, especially the language evolution seen through the lens of user-generated content, which gives access to a number of variants, socio- and idiolects. Methods consisting of "manually selecting, crawling and cleaning particular web sites with large and good-enough-quality textual content" (Spoustová and Spousta, 2012) are part of focused corpora, while focused crawling does not necessarily involve scrupulous work *a priori* but in any case the prioritization "towards documents which, according to some metric, have a high relevance" (Biemann et al., 2013). Even for comparatively large corpora, focused web corpus construction using pre-selected sources can lead to a higher yield and save time and resources while increasing the text quality of the resulting corpus (Schäfer et al., 2014).

The present use case concerns German, for which historical and contemporary corpora have been built as part of an aggregated lexical information platform (Geyken et al., 2017), the Digital Dictionary of the German Language (DWDS).[1] Specialized web corpora are built (Barbaresi, 2016) which can then be compared to existing resources such as newspaper and general-purpose corpora. Among other things, such corpora can be used to search for definitory elements related to newly created words or word senses (Barbaresi et al., 2018), for example by means of an automated content extraction and manual screening of pre-selected results.

A fundamental argument in favor of such corpora is related to the principles of the "Net economy" with the re-composition of the media landscape it fosters. It has seen the raise of "immaterial labor", "a social power that is independent and able to organize both its own work and its relations with business entities", where notions of "leisure time" and "working time" are fused and where the "split between author and audience" is transcended (Lazzarato, 1996). In some contexts the notion of "free labor" is also relevant to describe "the moment where [the] knowledgeable consumption of culture is translated into productive activities." (Terranova, 2000) These conditions of text production have to be accounted for, notably because they help creating a "long tail of bloggers who get little or no remuneration" (Rocamora, 2018) Community-building and content publishing among producers-consumers result in a major increase of text production which leads to more efficient corpus construction and potentially to a text collection that is easier to categorize.

Blogs seem to be particularly adequate as "the practice of blogging involves producing digital content with the intention of sharing it asynchronously with a conceptualized audience." (boyd, 2006) From the beginning of research on blogs/weblogs, the main definitory criterion has been their form, a reverse chronological sequences of dated entries and/or the use of dedicated software to articulate and publish the entries, a "weblog publishing software tool" (Glance et al., 2004) or content management system. Blogs are dynamic in nature, in consequence they "differ from static webpages because they capture ongoing expressions, not the edits of a static creation." (boyd, 2006) Another potential advantage in the case of focused crawls consists of the community-building aspects, as blogs are intricately intertwined in what has been called the blogosphere, as the active cross-linking helps to "create a strong sense of community" (Glance et al., 2004), which could help to find series of texts on a given topic by following links, that is by way of web crawling (Olston and Najork, 2010).

Difficulties raised by blogs as research objects are of conceptual and practical nature. First, the definition of what belongs to the genre and its use as a single category is controversial (Garden, 2012). This typology has notably been criticized for not being specific enough, especially concerning the sociolinguistic setting (Lomborg, 2009). A further demarcation can be made between blogs and social networks restricted to a single platform: "They differ from community tools because the expressions are captured locally, not in a shared common space." (boyd, 2006) These local spaces feature much less restrictions for machine-based access but also feature less directly exploitable

[1] https://www.dwds.de

metadata, although the profusion of user data on social media platforms can be of great value, for example to study linguistic variation (Barbaresi and Ruiz Tinoco, 2018). Consequently, the extraction of relevant content and metadata is highly relevant in order to make such web corpora exploitable. Finally, the commonly found term of blogosphere suggests a connection that does not necessarily exist, in opposition to the concept of "blogipelago", which "reminds us of separateness, disconnection, and the immense effort it can take to move from one island or network to another" (Dean, 2010). This effort clearly impacts corpus construction by requiring more screening as well as significant "island hopping". This is for example the case in communities which are fairly small and disconnected from other websites on the topic, e.g. Austrian fashion blogs which appear to refer to each other but do not often include links to other similar communities or topics. In the end, it is quite rare to find ready-made resources, especially for a topically focused approach, so that gathering methods and criteria ought to be discussed. As in genre-based studies, manual annotation – for example through crowdsourcing – can be an option for assessing the content of web texts and pave the way for classification tasks, but the lack of pre-existing data makes a pioneering work necessary (Asheghi et al., 2014). Provided this assumption is correct, collecting restricted portions of the Web for linguistic research remains nevertheless possible with sufficient screening.

3 Preliminary conclusions

Following the research on blogs/weblogs, we define blogs according to their form, consisting of dated entries available online and often managed by a broadly available publishing tool or web space. The discovery of relevant portions of the web is performed semi-automatically by pre-selecting hundreds of sources. Second, important metadata such as the publication date and main text content are extracted automatically based on structural patterns as well as heuristic criteria on text and markup. The resulting text base resides in a subset of web pages which have been found, downloaded and processed; documents with non-existent or missing date or entry content are discarded during processing and are not part of the corpus. By checking the seen web pages as to their relevance, it becomes possible to benefit from the insertion into a "web territory" (Cardon et al., 2011) that implies virtual communities as well as a complex adaptation process, which is also relevant from a linguistic standpoint. Surveys of particular portions of the web can also feature additional criteria such as content licensing, as some public licenses could help contributing back the corpus construction work to the research community.

We need both data and scientific instruments to shed light on subfields of the digital public sphere such as websites devoted to information technology (Pohlmann and Barbaresi, 2019), fashion & beauty, or literature. These topics in particular have the advantage of being among the most present online while mostly addressing complementary "prosumer" communities, even if studies relying on website publishing and blogging activities face a long tail with respect to impact and readership as well as concerning the move towards other publishing platforms and other content types. Nonetheless, some interlinking exists, webpages and especially blogs are still alive and relevant to gather corpus evidence. In the end, compared to "pre-web" and general-purpose corpora, challenges reside (1) in the necessity to consider texts types and topics beyond the previous extension of these notions and beyond known categories, (2) in a corresponding mapping of relevant portions of the web, and (3) in the ability to extract and pre-process resulting web texts and ultimately to make them available in clearly describable and coherent collections.

References

Noushin Rezapour Asheghi, Serge Sharoff, and Katja Markert. 2014. Designing and Evaluating a Reliable Corpus of Web Genres via Crowd-Sourcing. In *9th Language Resources and Evaluation Conference (LREC 2014)*, pages 1339–1346.

Adrien Barbaresi. 2015. *Ad hoc and general-purpose corpus construction from web sources*. Ph.D. thesis, École Normale Supérieure de Lyon.

Adrien Barbaresi. 2016. Efficient construction of metadata-enhanced web corpora. In *Proceedings of the 10th Web as Corpus Workshop*, pages 7–16. Association for Computational Linguistics.

Adrien Barbaresi, Lothar Lemnitzer, and Alexander Geyken. 2018. A database of German definitory contexts from selected web sources. In *11th International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3068–3073. European Language Resources Association (ELRA).

Adrien Barbaresi and Antonio Ruiz Tinoco. 2018. Using Elasticsearch for Linguistic Analysis of Tweets in Time and Space. In *Proceedings of the LREC 2018 Workshop CMLC-6*, pages 14–19. ELRA.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Marco Baroni and Motoko Ueyama. 2006. Building general- and special-purpose corpora by Web crawling. In *Proceedings of the 13th NIJL International Symposium, Language corpora: Their compilation and application*, pages 31–40.

Gunnar Bergh and Eros Zanchetta. 2008. Web linguistics. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics, An International Handbook*, pages 309–327. Mouton de Gruyter, Berlin.

Tim Berners-Lee, Wendy Hall, James A. Hendler, Kieron O'Hara, Nigel Shadbolt, and Daniel J. Weitzner. 2006. A Framework for Web Science. *Foundations and Trends in Web Science*, 1(1):1–130.

Chris Biemann, Felix Bildhauer, Stefan Evert, Dirk Goldhahn, Uwe Quasthoff, Roland Schäfer, Johannes Simon, Leonard Swiezinski, and Torsten Zesch. 2013. Scalable Construction of High-Quality Web Corpora. *Journal for Language Technology and Computational Linguistics*, pages 23–59.

danah boyd. 2006. A Blogger's Blog: Exploring the Definition of a Medium. *Reconstruction*, 6(4):1–21.

Dominique Cardon, Guilhem Fouetillou, and Camille Roth. 2011. Two Paths of Glory – Structural Positions and Trajectories of Websites within Their Topical Territory. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Jodi Dean. 2010. *Blog theory: Feedback and capture in the circuits of drive*. Polity, Cambridge.

Mary Garden. 2012. Defining blog: A fool's errand or a necessary undertaking. *Journalism*, 13(4):483–499.

Alexander Geyken, Adrien Barbaresi, Jörg Didakowski, Bryan Jurish, Frank Wiegand, and Lothar Lemnitzer. 2017. Die Korpusplattform des "Digitalen Wörterbuchs der deutschen Sprache" (DWDS). *Zeitschrift für germanistische Linguistik*, 45(2):327–344.

Natalie Glance, Matthew Hurst, and Takashi Tomokiyo. 2004. Blogpulse: Automated trend discovery for weblogs. In *WWW 2004 workshop on the weblogging ecosystem: Aggregation, analysis and dynamics*, volume 2004, New York.

Stefan T. Gries. 2009. What is Corpus Linguistics? *Language and Linguistics Compass*, 3(5):1225–1241.

James Hendler, Nigel Shadbolt, Wendy Hall, Tim Berners-Lee, and Daniel Weitzner. 2008. Web Science: An Interdisciplinary Approach to Understanding the Web. *Communications of the ACM*, 51(7):60–69.

Adam Kilgarriff. 2007. Googleology is bad science. *Computational Linguistics*, 33(1):147–151.

Maurizio Lazzarato. 1996. Immaterial Labor. In P. Virno and M. Hardy, editors, *Radical Thought in Italy*, pages 132–146. University of Minnesota Press.

Stine Lomborg. 2009. Navigating the blogosphere: Towards a genre-based typology of weblogs. *First Monday*, 14(5).

Anne O'Keeffe and Michael McCarthy. 2010. *The Routledge handbook of corpus linguistics*. Routledge.

Christopher Olston and Marc Najork. 2010. Web Crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246.

Jens Pohlmann and Adrien Barbaresi. 2019. Diving into the Conplexities of the Tech Blog Sphere. In *Digital Humanities 2019 Book of Abstracts*. ADHO.

Antoinette Renouf. 2007. Corpus development 25 years on: from super-corpus to cyber-corpus. In *Corpus Linguistics 25 years on*, pages 27–49. Brill Rodopi.

Agnès Rocamora. 2018. The labour of fashion blogging. In Leah Armstrong and Felice McDowell, editors, *Fashioning Professionals*. Bloomsbury.

Geoffrey Sampson. 2000. The role of taxonomy in language engineering. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1769):1339–1355.

Roland Schäfer, Adrien Barbaresi, and Felix Bildhauer. 2014. Focused Web Corpus Crawling. In *Proceedings of the 9th Web as Corpus workshop (WAC-9) @ EACL 2014*, pages 9–15. Association for Computational Linguistics.

Serge Sharoff. 2018. Functional text dimensions for the annotation of web corpora. *Corpora*, 13(1):65–95.

Johanka Spoustová and Miroslav Spousta. 2012. A High-Quality Web Corpus of Czech. In *Proceedings of LREC*, pages 311–315.

Tiziana Terranova. 2000. Free labor: Producing culture for the digital economy. *Social text*, 18(2):33–58.

Mark Warschauer and Douglas Grimes. 2007. Audience, authorship, and artifact: The emergent semiotics of web 2.0. *Annual Review of Applied Linguistics*, 27:1–23.

What's New in EuReCo?
Interoperability, Comparable Corpora, Licensing

Marc Kupietz, Eliza Margaretha, Nils Diewald, Harald Lüngen, Peter Fankhauser
Leibniz-Institut für Deutsche Sprache
R5 6–13, 68161 Mannheim, Germany
{kupietz|margaretha|diewald|luengen|fankhauser}@ids-mannheim.de

Abstract

This paper reports on the latest developments of the European Reference Corpus EuReCo and the German Reference Corpus in relation to three of the most important CMLC topics: interoperability, collaboration on corpus infrastructure building, and legal issues. Concerning interoperability, we present new ways to access DeReKo via KorAP on the API and on the plugin level. In addition we report about advancements in the EuReCo- and ICC-initiatives with the provision of comparable corpora, and about recent problems with license acquisitions and our solution approaches using an indemnification clause and model licenses that include scientific exploitation.

1 Interoperability

At the last CMLC workshop on the special topic of interoperability, we presented our general concept of how to make DeReKo data available as comprehensively and freely as possible, taking into account all legal restrictions (Kupietz et al., 2018). In this context, we had defined four levels at which an improvement of accessibility of corpus data was desirable and feasible:

1. data level
2. API level
3. plugin level
4. source code level

In the meantime, there have been relevant developments, especially with regard to the API level, about which we would like to report in the following section.

1.1 API Level

Authorization

The notorious problem with language resources as research data is that in the vast majority of cases, they are not free from the rights of third parties that do not belong to the scientific community (Kupietz et al., 2010). Thus, often the required rights of use have to be transferred from the right holder via a corpus provider to the end user by signing licence agreements. In the scenario of web corpus management tools, these agreements are then referred to, when a user is authenticated. In browsers this procedure is comparatively unproblematic, as the authorization can be handled in a common session management flow. For programmatic access to corpora via Web service APIs, the problem in the area of language resources, even in the context of the large CLARIN initiative, is, however, in practice unsolved. It is necessary to tackle some significant challenges including authentication of users and external user applications, as well as explicit authorization from users for the applications to access their data and the corpora on their behalf. In a complex scenario involving a system of multiple chains of independent applications like the CLARIN Federated Content Search, there is also an issue of delegating authorization from one application to another. In KorAP, to provide API access to DeReKo, we make use of the OAuth 2.0[1] protocol dealing particularly with authorization procedures.

Public Metadata Requests

A particularly simple approach to dealing with legal obstacles, which could even do without authentication and authorization, is to limit the disclosure of data to those that are not protected by copyright, provided that the origin and creation of such data is legitimate, e.g. through licensing agreements and/or copyright exceptions.

In the case of DeReKo, this approach can probably be used for a fairly broad range of applications, covering the analysis of frequency distributions in

[1] https://oauth.net/2/

relation to metadata variables such as publication date, publication location and subject area, thus including application scenarios such as diachronic analysis and comparison of language variants. In the context of the automatically processed part of such investigations, the output of textual, copyrighted data can usually be completely dispensed with; only the hits and their metadata are required.

KorAP manages user access to copyrighted and otherwise restricted data by using a query rewrite mechanism (Bański et al., 2014) that restricts access only to available resources according to user agreements and access location. Public metadata requests allow performing actions such as query search involving restricted data, but only the public metadata of each result are returned as output. The actual text snippet of the matches and non-public metadata are omitted. To provide public metadata requests of all resources to unauthenticated users, we introduce an additional request parameter `access-rewrite-disabled=true` allowing KorAP to disable this particular rewrite. Nonetheless, the rewrite is not disabled for requests requiring user authentication or authorization, such as requesting non-public metadata, requesting not sufficiently licensed corpora, and requesting metadata of virtual corpora restricted to a user or a group.

Listing 1 shows the JSON response to a simple query for the keyword 'Monnemer'. It comprises the generated operator tree for the `query`, the rewritten operator tree for the metadata constraints (`collection`), and the actual `matches`. As shown in Table 1, the current API provides only the metadata for every search hit. The advantage of such an unaggregated output is that it keeps the API simple and lets the user freely analyse any combination of metadata variables.

Listing 2 shows complete functions to query the DeReKo/KorAP API in R for (1) the size of the (virtual) corpus and (2) a search term. Note that the search function also provides a link to a corresponding albeit restricted request[3] to the KorAP web user interface (line 19), so that query results can be validated and analysed also manually. The result of a simple query for 'Hatespeech' is shown in Table 1. Apart from the support for multiple

[2] via http://korap.ids-mannheim.de/api/v1.0/search?ql=poliqarp&q=Monnemer&access-rewrite-disabled=true

[3] As usual, this requires a login – and the user to be authorized to access the requested data including the primary data.


```
{
  "@context": "http://korap.ids-mannheim.de/ns/
      KoralQuery/v0.3/context.jsonld",
  "meta": {
     "fields": ["textSigle","title","availability"],
     ...
  },
  "query": {
     "@type": "koral:token",
     "wrap": {
        "@type": "koral:term",
        "match": "match:eq",
        "layer": "orth",
        "key": "Monnemer",
        "foundry": "opennlp"
     }
  },
  "collection": {
     "operands": [{
           "@type": "koral:doc",
           "match": "match:eq",
           "type": "type:regex",
           "value": "CC-BY.*",
           "key": "availability"
        },{
           "operands": [{
                 "@type": "koral:doc",
                 "match": "match:eq",
                 "type": "type:regex",
                 "value": "ACA.*",
                 "key": "availability"
              },{
                 "operands": [{
                       "@type": "koral:doc",
                       "match": "match:eq",
                       "type": "type:regex",
                       "value": "QAO-NC",
                       "key": "availability"
                    },{
                       "@type": "koral:doc",
                       "match": "match:eq",
                       "type": "type:regex",
                       "value": "QAO.*",
                       "key": "availability"
                    }
                 ],
                 "@type": "koral:docGroup",
                 "operation": "operation:or"
              }
           ],
           "@type": "koral:docGroup",
           "operation": "operation:or"
        }
     ],
     "@type": "koral:docGroup",
     "operation": "operation:or",
     "rewrites": [{
        "@type": "koral:rewrite",
        "src": "Kustvakt",
        "operation": "operation:insertion",
        "scope": "availability(ALL)"
     }]
  },
  "matches": [
     {
        "matchID": "match-WDD17/M00/35548-p730-731",
        "textSigle": "WDD17/M00/35548",
        "availability": "CC-BY-SA",
        "title": "Diskussion:Mannheim"
     },{
        "matchID": "match-WDD17/M00/35548-p777-778",
        "textSigle": "WDD17/M00/35548",
        "availability": "CC-BY-SA",
        "title": "Diskussion:Mannheim"
     },{
        "matchID": "match-HMP18/FEB/00566-p153-154",
        "textSigle": "HMP18/FEB/00566",
        "availability": "QAO-NC",
        "title": "Der Rockstar unter den Comedians"
     }
  ]
}
```


Listing 1: Shortened JSON result of the query for 'Monnemer'[2].

```r
library(jsonlite)
korapurl <- "https://korap.ids-mannheim.de/"
apiurl <- paste0(korapurl, 'api/v1.0/')

fields <- c("corpusSigle", "textSigle", "pubDate", "pubPlace",
            "availability", "textClass")

derekoStats <- function(vc='') {
  return(fromJSON(paste0(apiurl, 'statistics?cq=',
                         URLencode(vc, reserved=TRUE))))
}

derekoQuery <- function(query, vc="", ql="poliqarp") {
  page <- 1
  results <- 0
  request <- paste0('?q=', URLencode(query, reserved=TRUE),
                    ifelse(vc != '', paste0('&cq=', URLencode(vc, reserved=TRUE)), ''),
                    '&ql=', ql);
  print(paste0("corresponding KorAP-UI request: ", paste0(korapurl, request)))
  repeat {
    res <- fromJSON(paste0(apiurl, 'search', request,
        '&count=50&fields=', paste(fields, collapse = ","),
        '&access-rewrite-disabled=true&page=', page))
    if (res$meta$totalResults == 0) { return(data.frame()) }
    for (field in fields) {
      if (!field %in% colnames(res$matches)) {
        res$matches[, field] <- NA
      }
    }
    currentMatches <- res$matches[fields]
    factorCols <- colnames(subset(currentMatches, select=-c(pubDate)))
    currentMatches[factorCols] <- lapply(currentMatches[factorCols], factor)
    currentMatches$pubDate = as.Date(currentMatches$pubDate, format = "%Y-%m-%d")
    if (page == 1) {
      allMatches <- currentMatches
      expectedResults <- res$meta$totalResults
    } else {
      allMatches <- rbind(allMatches, currentMatches)
    }
    print(paste0("Retrieved page: ", page, "/",
                 ceiling(expectedResults / res$meta$itemsPerPage)))
    page <- page + 1
    results <- results + res$meta$itemsPerPage
    if (results >= expectedResults) {
      break
    }
  }
  return(allMatches)
}
```

Listing 2: R sample functions to query the DeReKo / KorAP API. `derekoStats` returns the size of a (virtual) corpus and `derekoQuery` returns the results of a search for some term as a data frame.

textClass	textSigle	pubPlace	availability	pubDate	corpusSigle
staat-gesellschaft biographien-interviews	SOL13/SEP/01462	Hamburg	QAO-NC	2013-09-14	SOL13
politik ausland politik inland	T15/AUG/00332	Berlin	QAO-NC	2015-08-04	T15
politik inland staat-gesellschaft familie-geschlecht	S15/SEP/00251	Hamburg	QAO-NC	2015-09-19	S15
politik inland staat-gesellschaft familie-geschlecht	S15/SEP/00251	Hamburg	QAO-NC	2015-09-19	S15
politik inland staat-gesellschaft familie-geschlecht	S15/SEP/00251	Hamburg	QAO-NC	2015-09-19	S15
kultur literatur	SOL15/SEP/02745	Hamburg	QAO-NC	2015-09-30	SOL15
staat-gesellschaft familie-geschlecht	RHZ15/NOV/03331	Koblenz	QAO-NC	2015-11-05	RHZ15
staat-gesellschaft familie-geschlecht	RHZ15/NOV/03331	Koblenz	QAO-NC	2015-11-05	RHZ15
staat-gesellschaft familie-geschlecht wissenschaft populaerwissenschaft	T15/NOV/02335	Berlin	QAO-NC	2015-11-24	T15
technik-industrie edv-elektronik wissenschaft populaerwissenschaft	T15/DEZ/00520	Berlin	QAO-NC	2015-12-05	T15
politik inland	T15/DEZ/01762	Berlin	QAO-NC	2015-12-17	T15
politik inland	T15/DEZ/01762	Berlin	QAO-NC	2015-12-17	T15
politik inland	T15/DEZ/01762	Berlin	QAO-NC	2015-12-17	T15
politik inland	T15/DEZ/01762	Berlin	QAO-NC	2015-12-17	T15
staat-gesellschaft biographien-interviews	SOL16/JAN/01169	Hamburg	QAO-NC	2016-01-14	SOL16
...					

Table 1: Output of the first 15 results from `derekoQuery("Hatespeech")` using the R function from Listing 2, sorted by publication date.

query languages (Bingel and Diewald, 2015), the API also supports the restriction of searches to virtual sub-corpora based on metadata properties. A more complex example query, that involves a more complex search referring to multiple POS and lemma annotations as well as a virtual corpus definition is shown in Listing 3.

In the near future we will provide libraries to access the DeReKo/KorAP API for different programming languages, starting with R. In order to comply with license agreements and/or the § 60d UrhG text and data mining exception, the access will be limited to academic, non-commercial use.

A current and more detailed documentation of the API can be found in the Wiki of the KorAP component Kustvakt on KorAP's github page.[4]

1.2 Plugin Level

The KorAP user interface provides several entry points to embed results and configuration options for plugins (Diewald et al., to appear). Views can be embedded in so-called *panels* in the user interface, currently available for views on a) the virtual corpus, b) the search result, and c) matches. These entry points are still in an early stage and interactions with the user interface are initially limited. They are planned to be cautiously extended on demand, mainly for security reasons. For example, embedded plugins can already send messages to the global notification system of the user interface, but cannot alter query strings or virtual corpus definitions yet. In case a plugin requires access to the corpus data (for example to provide specific data visualisations), it can communicate via the API, authorized using OAuth 2.0. The first plugins under preparation focus on export capabilities embedded in the search result panel and communicate with the search API.

[4] https://github.com/KorAP/Kustvakt/wiki

2 Comparable Corpora

As discussed at the penultimate CMLC workshop, IDS participates in two essentially complementary initiatives to build comparable corpora: 1) the European Reference Corpus EuReCo (Kupietz et al., 2017) and 2) the International Comparable Corpus ICC (Kirk and Čermáková, 2017; Kirk et al., 2018).

2.1 EuReCo

Within the EuReCo initiave, the first pilot project *DRuKoLA* for the development of a German-Romanian corpus was completed in 2018 (Kupietz et al., to appear(a)). In this context, first virtual comparable corpora based on DeReKo and the Romanian reference corpus CoRoLa were defined and already used for first linguistic investigations (Kupietz et al., to appear(b)). In addition, parts of the Hungarian National Corpus were integrated into EuReCo framework within the 2nd pilot project *DeutUng* (Kupietz et al., to appear(b)).

```
> derekoQuery('[orth="[dw]as" & (tt/p=PRELS | opennlp/p=PRELS)] [tt/p=ADJA] [tt/l=
    sein]', 'corpusTitle="Der Spiegel" & pubDate since 2017-01-01')
[1] "Retrieved page: 1/1 (2.146951205 s)"
        textSigle    pubDate                                  textClass
1  S17/SEP/00243 2017-09-16 staat-gesellschaft biographien-interviews
2  S18/APR/00171 2018-04-14 staat-gesellschaft biographien-interviews
3  S18/MAR/00397 2018-03-24         wissenschaft populaerwissenschaft
4  S18/SEP/00396 2018-09-22 staat-gesellschaft biographien-interviews
5  S18/JUL/00234 2018-07-21                              politik inland
6  S17/AUG/00322 2017-08-26   biographien-interviews kultur literatur
7  S18/SEP/00156 2018-09-08                              politik inland
8  S18/AUG/00281 2018-08-18                              sport fussball
9  S17/MAI/00359 2017-05-27     staat-gesellschaft familie-geschlecht
10 S18/MAI/00069 2018-05-05              freizeit-unterhaltung reisen
11 S18/JUN/00010 2018-06-02                            politik ausland
12 S18/APR/00342 2018-04-28                           kultur literatur
13 S18/JAN/00285 2018-01-20                                kultur film
```

Listing 3: Complex query for 'das' (the/that) or 'was' (what) annotated as relative pronoun by TreeTagger or by the OpenNLP tools, followed by an attributive adjective and a form of 'sein' (to be), according to the TreeTagger annotations, in a virtual sub-corpus restricted to issues of the news magazine Der Spiegel published since 1st January 2017.

2.2 ICC

While EuReCo rather uses a primordial sample design approach (Kupietz et al., 2010) and wants to enable users to define a virtual comparable corpus based on the underlying individual language corpora, depending on the task and language domain investigated, the composition of the target corpus of the ICC initiative is determined from the outset to mimic the one of the *International Corpus of English* (ICE), with a few exceptions. The ICC plan is to complete at least the written linguistic corpus parts for some languages by 2019.

3 Legal issues and Licensing

The German reference corpus DeReKo relies on and continuously acquires licenses for the scientific use of text content, mostly from publishing companies. Many newspaper publishers are prepared to grant a free license for the use of their latest content in the DeReKo scenario (i. e. performing query and analysis via the dedicated corpus research interface that displays results only as text snippets in a KWIC format, or querying metadata and deriving statistics via the new KorAP API described above). Book publishers (both of fiction or specialised books), however, have on average not been not so generous, i. e. many do not reply to our acquisition campaigns in the first place, and those who do, grant licenses only for a limited number of titles most of the time. We attribute this firstly to some reluctance to make content available that is still actively being marketed, and secondly to the much higher need of time and effort to select and provide book content to external archives because it is simply not part of their established workflows (Kupietz and Lüngen, 2014). Since 2018, we have come across the new phenomenon that book publishers told us that they appreciated our project and would be willing to grant licenses, but they could not say whether they could grant rights for scientific use of their books in DeReKo, not being able to know whether they actually are in a legal position to do so. They would have to look into each particular author contract to assess this, which would be (too) costly to do (given that DeReKo would like to get the licenses for free). The risk of being sued for a breach of intellectual property rights by an author if they still granted us licenses was indeed considered low, however seemed not worth to be taken by the publisher if they have no gain from the deal. Another publisher had sought a legal opinion which stated that the type of use of their content in DeReKo was not at all covered by the model contract for authors provided by the Publishers and Booksellers Association that they generally use.

The main reason for these apparently new problems and the deterioration in the acquisition of books was that §§ 31a UrhG "Contracts for Unknown Types of Use" and 137l UrhG "Transitional Provisions for New Types of Use", which entered the German Copyright Act on 1 January 2008 and which essentially state that older author contracts automatically permit electronic exploitation unless

the author objects, initially made the acquisition of rights much easier. Subsequently, it seemed that publishers first reviewed their author contracts, including also newer ones that were no longer actually affected by the amendment, with regard to electronic rather than scientific exploitation. This seems to have changed by now.

As a first reaction to the problem, we have added a new indemnity clause against third party intellectual property claims to our standard agreements with the help of our legal experts. The idea is that the risk will be taken by the IDS, or that explicit licenses will subsequently be acquired directly from the authors. A second measure will be to approach the Publishers Association and ask them to explicitly include the scientific type of use of linguistic analysis in their model contract. In doing so, we hope to make book publishers more willing to grant free licenses for the scientific use of text content in the DeReKo scenario.

4 Conclusions

Apart from the reports on progress in the provision of comparable corpora in European languages and the long-term consequences of a 10-year-old amendment to the German Copyright Act, this paper has above all shown new ways in which, despite legal and ultimately economic hurdles, large corpora can be opened up for programmatic frequency analyses without infringing on the interests or rights of right holders and without incurring great technical expense.

The method we have presented here and implemented for DeReKo and KorAP basically follows our motto borrowed from Jim Gray (2003):

> *If the data is to big or not allowed to move, put the computation near the data.* (cf. Kupietz et al., 2010, 2014, 2018)

with the addendum:

> *If not all computation can be put near the data, move just such data that is allowed and required to move.*

References

Piotr Bański, Nils Diewald, Michael Hanl, Marc Kupietz, and Andreas Witt. 2014. Access Control by Query Rewriting: the Case of KorAP. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik. European Language Resources Association (ELRA).

Joachim Bingel and Nils Diewald. 2015. KoralQuery – a General Corpus Query Protocol. In *Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools at NODALIDA 2015*, Vilnius, Lithuania.

Nils Diewald, Verginica Barbu Mititelu, and Marc Kupietz. to appear. The KorAP user interface. Accessing CoRoLa via KorAP. *Revue Roumaine de Linguistique*.

Jim Gray. 2003. Distributed Computing Economics. Technical Report MSR-TR-2003-24, Microsoft Research.

John Kirk and Anna Čermáková. 2017. From ICE to ICC: The new International Comparable Corpus. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora and Big Data and Natural Language Processing (CMLC-5+BigNLP) 2017*, pages 7 – 12. IDS. https://ids-pub.bsz-bw.de/frontdoor/index/index/docId/6249.

John Kirk, Anna Čermáková, Signe Oksefjell Ebeling, Jarle Ebeling, Michal Kren, Karin Aijmer, Vladimir Benko, Radovan Garabik, Rafal Gorski, Jarmo Jantunen, Marc Kupietz, Maria Simkova, Thomas Schmidt, and Oliver Wicher. 2018. Introducing the International Comparable Corpus. In *Book of Abstracts. Using Corpora in Contrastive and Translation Studies Conference (5th edition), CECL Papers 1*, pages 96 – 97, Louvain-la-Neuve. Université catholique de Louvain.

Marc Kupietz, Cyril Belica, Holger Keibel, and Andreas Witt. 2010. The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, page 1848–1854, Valletta, Malta. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2010/pdf/414_Paper.pdf.

Marc Kupietz, Anda Cosma, and Andreas Witt. to appear(a). The DRuKoLA project. *Revue Roumaine de Linguistique*.

Marc Kupietz, Ruxandra Cosma, Dan Cristea, Nils Diewald, Beata Trawinski, Dan Tufis, Tamás Váradi, and Angelika Wöllstein. to appear(b). Recent developments in the European Reference Corpus (EuReCo). In *Proceedings of UCCTS 2018*, Louvain-la-Neuve.

Marc Kupietz, Nils Diewald, and Peter Fankhauser. 2018. How to get the computation near the data: improving data accessibility to, and reusability of analysis functions in corpus query platforms. In *Proceedings of the LREC 2018 Workshop "Challenges in the Management of Large Corpora (CMLC-6)" 07 May 2018 – Miyazaki, Japan*, pages 20 – 25, Paris. European language resources association (ELRA).

Marc Kupietz and Harald Lüngen. 2014. Recent developments in DeReKo. In *Proceedings of the ninth conference on international language resources and evaluation (LREC'14)*, pages 2378–2385, Reykjavik, Iceland. ELRA.

Marc Kupietz, Harald Lüngen, Piotr Bański, and Cyril Belica. 2014. Maximizing the potential of very large corpora: 50 years of big language data at IDS Mannheim. In *Proceedinigs of the LREC-2014-workshop challenges in the management of large corpora (CMLC2)*, pages 1 – 6, Reykjavik / Paris. ELRA.

Marc Kupietz, Andreas Witt, Piotr Bański, Dan Tufiş, Dan Cristea, and Tamás Váradi. 2017. EuReCo - Joining Forces for a European Reference Corpus as a sustainable base for cross-linguistic research. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora and Big Data and Natural Language Processing (CMLC-5+BigNLP) 2017*, pages 15 – 19. IDS. https://ids-pub.bsz-bw.de/frontdoor/index/index/docId/6258.