# The Aranea Corpora Family: Ten+ Years of Processing Web-Crawled Data

Vladimír Benko[1,2][0000−0002−4600−5515]

[1] Slovak Academy of Sciences, Ľ. Štúr Institute of Linguistics
Panská 26, 811 011 Bratislava, Slovakia
[2] Comenius University Science Park
UNESCO Chair in Plurilingual and Multicultural Communications
Ilkovičova 8, 814 04 Bratislava, Slovakia
vladimir.benko@juls.savba.sk
https://juls.savba.sk/~vladob

**Abstract** *Aranea* is a project to create a family of web-crawled corpora for languages taught at Slovak Universities. Since 2013, more than two dozen languages have been added to the project: they are often represented with Gigaword+ size corpora and sometimes have subcorpora for their territorial varieties. Our paper summarizes the development of the *Aranea* project in the past decade. We describe the step-by-step optimization of the processing pipeline, highlight existing issues, and discuss the linguistic rationale behind some engineering decisions associated with the idiosyncrasies of individual languages.

**Keywords:** web-crawled corpora · 'Brno pipeline' · FLOSS tools · ensemble lemmatization and PoS tagging.

## 1 Introduction

*Aranea*[3] is a multilingual corpus-building and corpus-managing framework designed to build web-corpora for languages taught at Slovak universities and primarily intended for pedagogical purposes. As the corpora are created using the same technology and have the same size(s), they can be considered "comparable" to a considerable extent[4].

While the project was officially released at the TSD conference in 2014 [5], this jubilee contribution is focused on the aspects of *Aranea* development that did not receive any or enough coverage elsewhere.

## 2 Related Work

Though the methodology and technology of building web-crawled corpora have been available for more than two decades, there have been just a few projects that used them to produce and maintain large multilingual collections.

[3] araneum *pl.* aranea *n.* (https://www.latin-dictionary.net/search/latin/araneum)
[4] The question of comparability with respect to web-crawled corpora is discussed in [7]

The *Web as Corpus* (*WaC*) project was a pioneer in this area, producing Gigaword corpora for four languages, i.e., *ukWaC* (British English), *deWaC*, *frWaC*, and *itWaC* [2,3], respectively.

The *TenTen* corpora collection is another notable example of a large-scale multilingual project that has created web-corpora for more than 40 languages and is making them available through the *Sketch Engine*[5] portal [11].

The {*bs,hr,sr*}*WaC* corpora resulted from using the same methodology and tools [14], while the *COW* (*Corpora of the Web*) yielded corpora for six languages using a different set of tools [22].

And finally, there is an impressive collection of corpora[6] created and maintained by Mark Davies from the Brigham Young University [9].

## 3 The *Aranea* Language Collection

The original intention of the *Aranea* project was to create corpora for Slovak, Czech and the four "large" languages (i.e. English, French, German, and Russian). However, over time our ambition first extended to covering all the languages taught at the Slovak Universities. Then, for various reasons, we have included other languages with available PoS taggers, even though these corpora are unlikely to be used by Slovak students or teachers in the foreseeable future.

## 4 Crawling and Pre-Tokenization Filtration

Since its publication in 2013, we have been using *SpiderLing*[7] [27,5,8] as the tool for crawling the web to obtain and pre-process the corpus data. From the initial Version 0.72 up to today's Version 2.2, *SpiderLing* has gradually developed into a mature, stable, and extremely efficient tool capable (within our IT infrastructure and setup) of gathering as much as two billion tokens of corpus data in 24 hours. During these recent years, important improvements have included reducing the RAM necessary to store the internal data structures, refining the language identification procedure, and providing more metadata for the downloaded documents (for example, the web page title, original and detected character encoding, language detected through the *CLD2*[8] procedure, etc.).

After the data for a language is downloaded, we apply the following stack of operations: deduplication, secondary language detection, and deletion of texts with encoding issues.

[5] https://www.sketchengine.eu/
[6] https://www.english-corpora.org/
[7] https://corpus.tools/wiki/SpiderLing
[8] https://github.com/CLD2Owners/cld2

### 4.1 Secondary and Tertiary Deduplication

Though *SpiderLing* is performing basic detection and removal of 100%-duplicate documents on the fly, there are situations when duplicate documents do appear in the output, for example, when the new crawling session had to be (for various reasons) launched without taking into consideration the "context" of the previous one. The deduplication procedure is based on Rabin's fingerprint method [20] which compares the cryptographic MD4 Message-Digest Algorithm[9] hashes calculated for every document.

We also delete identical subsequent paragraphs that often appear in the documents. Though it can be argued that we lose some interesting information by deleting those copies, we decided to delete them as we suspect that in most cases duplicate subsequent paragraphs are the result of search engine optimization (SEO) efforts. The removal is performed using the *uniq* program. An excerpt from a deduplication log (of the Norwegian data) is shown in Fig. 1. The left-hand column indicates the number of repeating paragraphs.

```
 3 <p heading="yes">Skolelaboratoriet</p>
 2 <p>Er du sikker på at du vil slette?</p>
 2 <p heading="yes">Vi inkluderer personer med
     utviklingshemming i det ordinære arbeidslivet</p>
 2 <p>Varen er bestilt fra leverandør, men leveringsdato er
     ikke bekreftet.</p>
 2 <p heading="yes">Løse Dine Sinus problemer</p>
10 <p>Læs, hvordan kunder verden over drager nytte af vores
    alsidige løsninger.</p>
```

**Figure 1.** Secondary (subsequent paragraph) deduplication in *Araneum Norvegicum*.

### 4.2 Secondary Language Detection

Although the language identification procedure works fairly well for detecting "sufficiently different" languages, it often fails for the closely related ones. We, therefore, apply a more sensitive procedure that uses character frequencies: for each language, we count characters appearing in that language and not in any of the others which are similar. For example, Czech can be identified by the presence of the *ě*, *ř*, and *ů* characters; Slovak by *ä*, *ĺ*, *ľ*, *ô*, and *ŕ*; Hungarian by *ő*, and *ű*, etc. The respective thresholds for removing documents in languages similar to the expected language of the crawled country domain are set experimentally.

### 4.3 Character Encoding Issues

It sometimes happens that a document is garbled, either as a whole by incorrect detection of the character encoding by *SpiderLing*, or partially, when the issue is

[9] https://datatracker.ietf.org/doc/html/rfc1186

associated only with some turns in online discussions (most likely due to a misunderstanding of encoding between the portal and the respective user browser). As a result, the text can contain artifacts like this: "*â€œYouâ€™re no piker!*". Here, the UTF-8 encoding was incorrectly identified as 8-bit, resulting in the garbled single quote signs. A series of regular expressions has been written to identify these issues.

## 5 Tokenization and Post-Tokenization Processing

Tokenization is a technical process, yet some linguistically motivated decisions have to be made here. We prefer having the same "tokenization policy" for all languages, even in situations when the tagger expects a slightly different input format.

We use the *Unitok*[10] [16] universal tokenizer with a custom parameter file provided for each language. The tokenization strategy mostly adheres to that defined by the default parameter files, with some additional modifications related to the definition of period-final abbreviations, rare clitics, and other idiosyncrasies of the respective languages.

### 5.1 Sentence Segmentation

A rule-based algorithm was written to segment the tokenized text into sentences. The algorithm relies on punctuation and period-final abbreviations already recognized by the tokenizer. The outcomes are reasonably high quality for languages using scripts that distinguish between upper and lowercase letters (Latin and Cyrillic); for the other languages, there is more noise in the results.

### 5.2 Detection and Removal of Partially Duplicate Documents

At this step in the corpus preprocessing, we again use a tool from the "Brno" pipeline – a program referred to as *Onion*[11] [19]. As it expects the corpus text to be already tokenized, it is best to use *Onion* at this step in the workflow.

The description of the process and the discussion related to the optimal parameters can be found in [4]. We would only mention here that (for all corpora) we use the same input parameters: an n-gram length of 5 and a similarity threshold value of 0.9.

Depending on the language, the proportion of deleted material can vary from 30% to as much as 60% – the higher value is usually due to unnecessary long crawling sessions that return too many similar downloaded documents.

[10] https://corpus.tools/wiki/Unitok
[11] https://corpus.tools/wiki/Onion

### 5.3 Comparing the Corpora

After the pre-processing, tokenization and deduplication are completed, the sizes of the corpus data for the *Aranea* languages can be compared. For most languages, we performed two crawls – the initial one and the second one in 2021 in an attempt to cover the COVID pandemic discourse. For some languages, the more recent downloads are still being processed. Table 1 reflects the corpora sizes after preprocessing by language and indicates the years of the available crawls.

## 6 Some Language Idiosyncrasies

### 6.1 Chinese, Clitics and Sub-Word Tokenization

Unlike most other languages, Chinese does not separate words by spaces, and a whole paragraph appears as a single string in the text. "Tokenization" is therefore performed by a heuristic algorithm that tries to guess the word boundaries. We used the program designed by Serge Sharoff [12].

Correct tokenization of clitics is crucial for languages such as English and French: it helps the tagger recognize expressions like *I haven't been* or *J'en parle*. This, however, means that the tokenization of clitics has to be compatible with what is expected by the tagger.

However, there are situations when the tagger is not capable of recognizing either a split or a joint clitic — this has to be treated for each language individually.

Sub-word tokenization is traditionally considered in some languages, notably Polish and Georgian. We, however, decided not to perform it within the framework of our project [13].

### 6.2 Digraphia

Two languages in our collection use two scripts (Cyrillic and Latin) in parallel. While in Serbia, the existence of digraphia is an official state policy, in Uzbekistan, the situation of a de facto digraphia may be considered as a result of the partially unsuccessful transition from Cyrillic to Latin at the beginning of the 1990s.

As in both cases, the respective taggers expect the texts to use the Latin script only; corpus linguistics projects usually convert everything to Latin, thus hiding the information about the original appearance of the respective texts. Contrary to that, we consider the presence of Cyrillic script an interesting sociolinguistic attribute and keep all the texts in their original scripts. We introduced a "normalized word form" as an additional attribute to be used for

[12] Serge's web page containing his language resources is currently not accessible.

[13] Sub-word tokenization is a rather complex topic that is beyond the scope of this paper.

**Table 1.** Available crawls and corpora sizes (in millions of tokens) after preprocessing by language (as of June 2024).

| Lang | Year(s) crawled | Tokenization & depuplication | |
|---|---|---|---|
| | | Last year processed | Size |
| ar | 2017, 2021 | 2017 | 172 |
| be | 2021, 2023 | 2023 | 483 |
| bg | 2016, 2021 | 2016 | 2,325 |
| cs | 2013, 2015–2018, 2021, 2022, 2024 | 2024 | 12,248 |
| da | 2023 | 2023 | 2,759 |
| de | 2013–2015, 2018, 2021 | 2018 | 9,555 |
| en | 2013–2015, 2017, 2021, 2024 | 2021 | 21,655 |
| es | 2013, 2014, 2021 | 2021 | 5,620 |
| et | 2017–2019, 2021 | 2019 | 3,553 |
| fa | 2022 | 2022 | 3,892 |
| fi | 2014, 2021 | 2014 | 1,629 |
| fr | 2013, 2015–2019, 2021, 2024 | 2024 | 20,915 |
| hu | 2014, 2021 | 2023 | 3,352 |
| hr | 2021 | 2021 | 1,886 |
| it | 2014, 2021 | 2014 | 2,001 |
| ja | 2021 | – | n/a |
| ka | 2014, 2018, 2021, 2023, 2024 | 2024 | 1,568 |
| kk | 2021 | 2021 | 340 |
| ko | 2021 | 2021 | 1,113 |
| la | 2018, 2020 | 2020 | 126 |
| lv | 2017, 2018, 2021 | 2018 | 864 |
| mt | 2017, 2021 | 2021 | 24 |
| nl | 2013, 2021 | 2015 | 1,592 |
| no | 2019, 2021–2023 | 2023 | 3,532 |
| pl | 2013, 2021–2023 | 2024 | 7,867 |
| pt | 2015, 2021 | 2021 | 1,683 |
| ro | 2017, 2021 | 2018 | 1,874 |
| ru | 2013–2018, 2019–2023 | 2021 | 43,800 |
| sk | 2013–2024 | 2024 | 8,711 |
| sl | 2021 | 2021 | 1,211 |
| sq | 2021 | 2021 | 294 |
| sr | 2021 | 2024 | 2,423 |
| sv | 2017, 2021 | 2017 | 1,955 |
| tr | 2023 | 2023 | 1,933 |
| tt | 2018, 2021 | 2021 | 152 |
| uk | 2014, 2015, 2021, 2022 | 2022 | 5,249 |
| uz | 2020, 2024 | 2024 | 1,712 |
| zh | 2015, 2021, 2024 | 2015 | 1,254 |

lemmatization and PoS tagging. Using the appropriate attribute, queries can be performed in Latin script (to get all occurrences) or in Cyrillic script (to get only the relevant ones).

### 6.3 Romanization

We expect users of languages with Latin and/or Cyrillic scripts to have a keyboard with the respective layout. For other scripts it may be useful to be able to perform queries transliterated to Latin. In our corpus collection, the latter applies to Georgian, Persian, and Korean, where a solution similar to the digraphia languages is used. Unlike Serbian and Uzbek, however, "Romanization" is applied not only to word forms but also to lemmas, so that all queries can be performed in both native and Romanized forms. The result of such a query in the Georgian corpus is shown in Fig. 2.

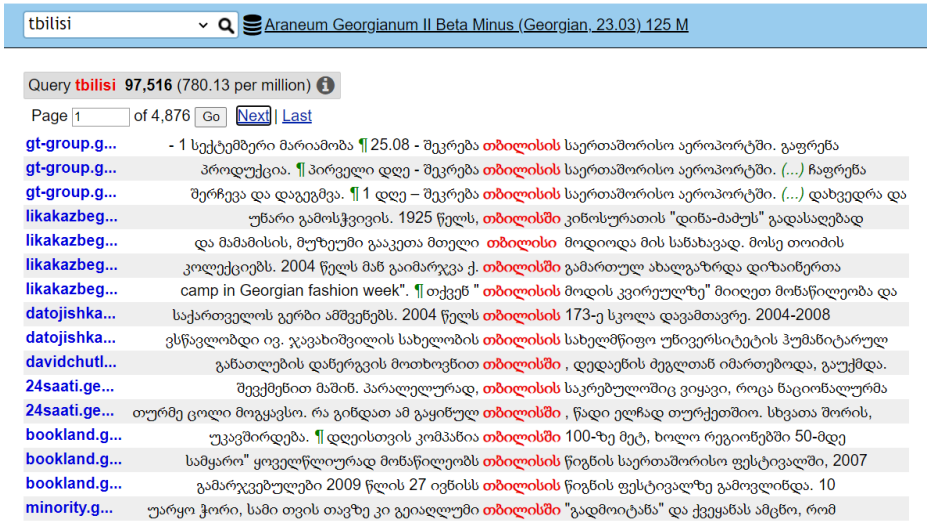


**Figure 2.** Querying for "tbilisi" in *Araneum Georgianum*.

### 6.4 "Half-Spaces"

One of the peculiarities of the standard Persian orthography is that some affixes are to be separated from their respective main word by a "zero-width non-joiner" character (U+200C) which prevents the neighboring characters from being joined into a ligature (as they would normally be). While this character can be entered on standard Persian keyboard, (apparently), in situations when writing on a different keyboard (such as Arabic), those affixes are either joined with the respective word or separated by a regular whitespace character. We provide

a special normalized attribute for "half-spaces", which makes querying more deterministic.

## 7 Lemmatization and PoS Tagging

### 7.1 The Baseline Annotation

For basic processing, we prefer tools with multilingual support, i.e., those equipped with language models for more than one language. Within the context of our project, we mostly use four taggers and one lemmatizer, as follows.

*TreeTagger*[14] [23] belongs to the oldest tools available and still being supported by its author, providing models for almost 50 languages. It is especially suitable for coarse tagsets, i.e., for languages with 'poor' morphology. Its main advantage is the tagging speed, sometimes more than by an order of magnitude higher than for newer neural networks-based tools. Its main deficiencies include lower precision for languages with rich morphology (Slavic and agglutinative languages) and no lemma guessing for the out-of-vocabulary (OOV) lexical items. The tag guessing, however, works reasonably well.

*UDPipe*[15] [25] is a tagger developed within the *Universal Dependencies*[16] (*UD*) [15] framework which provides language models for all languages with a suitably-sized dependency treebank. As those models rely on rules derived from the treebank data only, i.e. no additional morphological lexicon is considered, lemma guessing is performed for all lexical items, even for those present in the training data, with no testing whether the lemma was attested by the lexicon. At the time of writing, all *UD* annotations for the *Aranea* corpora were generated using *UDPipe 1* with the latest *UDs 2.5* language models released in December 2019. In the future, we plan to make use of the more recent version of the tagger, whose efficiency can be facilitated if graphic cards are used.

*MorphoDiTa*[17] [24] is a high-performance tagger combining, from the user perspective, functionalities of the two taggers mentioned above. However, it offers language models for just three languages (Czech, English, and Slovak) and is not supported anymore.

*Apertium*[18] [13] is an open-source machine translation system with a morphological module that can also be used separately for lemmatization and PoS tagging. Its main advantage is that it has language models for languages missing in other tools.

Unlike other tools mentioned above, *CSTlemma*[19] [12] performs lemmatization only, though it can consider PoS tags provided by other tools during disambiguation.

[14] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
[15] https://ufal.mff.cuni.cz/udpipe/1
[16] https://universaldependencies.org/
[17] https://ufal.mff.cuni.cz/morphodita
[18] https://www.apertium.org/
[19] https://cst.dk/online/lemmatiser/uk/

Two other tools used in our project include *LU MII Tagger*[20] for Latvian [17] and *HunPos*[21] [10] for Hungarian[22].

The behavior of these tools with respect to the OOV lexical items is summarized in Table 2.

**Table 2.** Baseline tools used for lemmatization and PoS tagging.

| Tool | | Non-OOV lexical items | OOV lexical items |
|---|---|---|---|
| TreeTagger | Lemmas | assigned from the lexicon | marked as 'unknown' |
| | PoS Tags | assigned from the lexicon | guessed |
| | Disambiguation | statistical | statistical |
| MorphoDiTa | Lemmas | assigned from the lexicon | guessed & marked |
| | PoS Tags | assigned from the lexicon | guessed & marked |
| | Disambiguation | statistical | statistical |
| UDPipe | Lemmas | n/a | guessed |
| | PoS Tags | n/a | guessed |
| | Disambiguation | n/a | statistical |
| Apertium | Lemmas | assigned from the lexicon | marked as 'unknown' |
| | PoS Tags | assigned from the lexicon | marked as 'unknown' |
| | Disambiguation | rule-based | n/a |
| CSTlemma | Lemmas | assigned from the lexicon[23] | guessed |
| | PoS Tags | n/a | n/a |
| | Disambiguation | statistical | statistical |

Each of the tools has pros and cons, and the decision to use a tool largely depends on the availability of a model for a specific language. In addition to the varying extent of language support, the tools also differ in processing speed – *UDPipe*, for example, is typically at least ten times slower than *TreeTagger*.

One of the measures to assess the quality of a language model is the lexical coverage measured by the OOV rate. For languages where the baseline tagger provides such information, the results are summarized in Table 3.

It can be seen that the lexical coverage varies a lot across languages – while being surprisingly high for some languages, it indicates questionable quality for other languages. To counteract this disparity, we applied an "ensemble" approach to lemmatization, and partially also to PoS tagging.

### 7.2 Ensemble Annotation

In Computational Linguistics, the "ensemble" term is used to describe approaches where several tools are used to (independently) perform the same operation, as-

[20] https://peteris.rocks/blog/latvian-part-of-speech-tagging/

[21] http://mokk.bme.hu/en/resources/hunpos/

[22] Hungarian was the only language with processing being "outsourced" – it has been done for us by the Research Institute of Linguistics at the Hungarian Academy of Sciences.

**Table 3.** Out-of-vocabulary (OOV) rates.

| Lang | Baseline tagger | Year tagged | OOVs (%) | Lang | Baseline tagger | Year tagged | OOVs (%) |
|---|---|---|---|---|---|---|---|
| ar | (not tagged yet) | | | la | TreeTagger | 2018 | 7.02 |
| be | TreeTagger | 2023 | 16.96 | lv | LU MII Tagger | 2018 | n/a |
| bg | TreeTagger | 2016 | 6.41 | mt | Apertium | 2021 | 16.28 |
| cs | MophoDiTa | 2020 | 2.50 | nl | TreeTagger | 2015 | 7.07 |
| da | TreeTagger | 2023 | 3.64 | no | TreeTagger | 2021 | 6.51 |
| de | TreeTagger | 2020 | 5.66 | pl | TreeTagger | 2021 | 9.47 |
| en | TreeTagger | 2017 | 2.72 | pt | TreeTagger | 2015 | 6.11 |
| es | TreeTagger | 2015 | 5.74 | ro | TreeTagger | 2019 | 5.97 |
| et | TreeTagger | 2019 | 17.50 | ru | TreeTagger | 2023 | 5.69 |
| fa | TreeTagger | 2022 | 10.48 | sk | MorphoDiTa | 2024 | 4.33 |
| fi | TreeTagger | 2015 | 12.38 | sl | TreeTagger | 2021 | 3.73 |
| fr | TreeTagger | 2021 | 4.26 | sq | Apertium | 2021 | 26.87 |
| hr | UDPipe | 2021 | n/a | sr | (not tagged yet) | | |
| hu | HunPos | 2014 | 2.46 | sv | TreeTagger | 2017 | 19.27 |
| it | TreeTagger | 2021 | 3.89 | tr | UDPipe | 2023 | n/a |
| ja | (not tagged yet) | | | tt | Apertium | 2021 | 6.79 |
| ka | TreeTagger | 2023 | 34.06 | uk | UDPipe | 2022 | n/a |
| kk | Apertium | 2021 | 6.03 | uz | Apertium | 2024 | 11.22 |
| ko | TreeTagger | 2021 | 14.16 | zh | TreeTagger | 2015 | 6.55 |

suming that aggregation of their outputs could improve the overall success rate of the whole process. In the framework of morphosyntactic annotation, we can speak about "ensemble lemmatization/tagging" if more than one lemmatizer/-tagger is available for a particular language, which is the case of several languages in our corpus collection.

If more than two taggers use the same tagset, the aggregation can be performed by simple majority voting. In our case, however, at least three "full-fledged" taggers are unavailable, and the respective tagsets are incompatible.

We started to apply the ensemble approach to the newer *Aranea* corpora created after 2022 and to the older corpora re-processed after this year. A typical setup consists of *TreeTagger*, *UDPipe* and *CSTlemma*. The main idea is to improve the annotation of the OOV items by "double guessing" (*UDPipe* + *CSTlemma* for lemmatization, and *TreeTagger* + *UDPipe* for PoS tags). Due to the differences in tagsets, we only consider main word classes encoded using the *Araneum Universal Tagset*[24] (*AUT*). The aggregation is rule-based, and can be described as follows:

1. If PoS can be assigned by means of a regular expression (punctuation, digits, symbols, e-mail addresses, etc.), ignore information provided by the taggers.
2. If the respective token was present in the morphological lexicon of *TreeTagger*, take both lemma and PoS from it.

[24] http://aranea.juls.savba.sk/aranea_about/aut.html

3. If the token was OOV in *TreeTagger* and *UDPipe* guessed a lemma, take both lemma and PoS from *UDPipe*. If the lemma guessed by *CSTlemma* is different, add it as an alternative. If PoS guessed by the *TreeTagger* is different, add it as an alternative.
4. Otherwise take the lemma from *CSTlemma* and PoS from *TreeTagger* and *UDPipe* (if different).

The result of the aggregation is stored in a special attribute "ztag": the respective value consists of two parts separated by a period – the left part has details about the assignment of lemma, while the right – about the PoS. The uppercase letters indicate success in the morphological lexicon lookup (in the case of *TreeTagger*), the lowercase letters indicate guessing, and the exclamation mark indicates that the respective value differs from that on the left. The actual situation in a 125-Megatoken (Farsi) *Araneum Persicum* is shown in Table 4 [6].

**Table 4.** Result of aggregation for *Araneum Persicum Minus* (125 million tokens)

| ztag | freq | % | ztag | freq | % |
|---|---|---|---|---|---|
| T!c.T!u | 313,397 | 0.25 | u!c.t!u | 734,753 | 0.59 |
| T!c.Tu | 1,744,755 | 1.40 | u!c.tu | 2,098,439 | 1.68 |
| T!u!c.T!u | 141,740 | 0.11 | uc.t!u | 2,077,455 | 1.66 |
| T!u!c.Tu | 2,082,343 | 1.67 | uc.tu | 3,751,262 | 3.00 |
| T!uc.T!u | 644,554 | 0.52 | c.t!u | 1,411,739 | 1.13 |
| T!uc.Tu | 4,477,277 | 3.58 | c.tu | 3,026,913 | 2.42 |
| Tc!u.T!u | 783,445 | 0.63 | z! | 6,629,046 | 5.30 |
| Tc!u.Tu | 1,242,515 | 0.99 | z# | 788,956 | 0.63 |
| Tc.T!u | 694,332 | 0.56 | z$ | 910,190 | 0.73 |
| Tc.Tu | 3,817,999 | 3.05 | z@ | 1,068 | < 0.01 |
| Tu!c.T!u | 455,065 | 0.36 | zu | 5,987 | < 0.01 |
| Tu!c.Tu | 13,464,342 | 10.77 | zv | 17,675 | 0.01 |
| Tuc.T!u | 7,834,406 | 6.27 | zw | 1,972 | < 0.01 |
| Tuc.Tu | 65,848,758 | 52.68 | Total | 125,000,383 | 100.00 |

Flags starting with the "z" letter indicate lexical items "tagged" by regular expressions. For example, "z!" denotes punctuation, "z#" numbers, "z$" symbols (special graphic characters, emoji, etc.), "z@" e-mail addresses, and "zu", "zv", and "zw" Internet addresses in different formats. As it can be seen, most items have a "Tuc.Tu" flag, indicating equal values assigned both for lemma and PoS by all tools, followed by a "Tu!c.Tu", where *CSTlemma* assigned a different lemma than *TreeTagger* and *UDPipe*. If we add counts for the "uc.tu" and "uc.t!" ztag values, we will find out that for 4.66% of the word forms both guessers assigned the same lemma, making it probably correct, and improving the overall lemmatization success rate.

## 8 Post-Tagging Processing

Lemmatization and PoS tagging result in corpus data that is minimally sufficient for some applications. However, for lexicographic purposes or language teaching, where the reading of concordances is often involved, the corpora must undergo additional processing steps.

### 8.1 Paragraph and Sentence-Level Deduplication

Detection and removal of partially duplicate paragraphs is performed by the *Onion* program (with the same settings as for the document deduplication, i.e. n-gram length of 5, and threshold level 0.9), and Rabin's fingerprint method is used for the sentences while ignoring numbers, special characters and punctuation (making use of the PoS annotation already present in the data). Both processes are described in [4].

The paragraph and sentence deduplication procedure typically removes 15–20% and 3–5% tokens, respectively, on top of the document deduplication.

### 8.2 Sentence-Level Language Identification

Foreign language (mostly English) sentences appear in almost every corpus of the *Aranea* family. They may be undesirable when frequency statistics are needed because frequent foreign words, such as English "the", can rank high in the frequency lists and contaminate the results for the targeted language.

Recently, we started removing foreign sentences using a procedure designed by Vít Suchomel [26] but based on our own implementation and on the statistics derived from the *Aranea* corpora. This process usually removes up to 5% of corpus tokens.

### 8.3 Removal of Spam and Other Machine-Generated Contents

There is no single method of detecting machine-generated content that often contains nonsensical parts. Until recently, auto-generated texts were relatively easy to identify using heuristic algorithms. With the advent of generative AI, texts produced by a machine are often indistinguishable from those created by a human.

This step appears as the last step in the corpus processing pipeline because it is relatively independent of the previous stages. Given a method for identification of "suspicious" documents, the preprocessed corpus can be put through a filtering procedure to remove spam and machine-generated content.

As spotting problematic texts depends on comprehension of the respective language, we were able to apply the large-scale spam-removing procedures to the Czech, Slovak, and Russian data only. The Czech and Slovak spam pages were identified as "link farms" advertising online bets, while Russian Internet spam was associated with prostitution. In both cases, the activities are illegal in the respective countries.

## 9 Processing by the Corpus Manager

To keep our corpora as comparable as possible, we sample them into two sizes: 125 million token corpora for teaching purposes and (if possible) 1.25 billion token corpora for general use. For some languages, we also have the as-much-as-can-get size.

Our two *Aranea* corpus portals have several thousand registered users, over two hundred of whom access our corpora online daily. On demand, we also provide processed source corpus data (for research purposes).

A final note on the last processing step in the preparation of a corpus is associated with the corpus manager. We use the *NoSketch Engine* [21], an open-source subset of the commercial *Sketch Engine*. To our knowledge, this is the only system with a FLOSS license capable of processing corpora of several billions of tokens in size.

One of the functionalities less known to users is the possibility of creating user sub-corpora, either using the metadata or by saving the result of any corpus query. Such sub-corpora can subsequently be used to calculate keywords, for example, by comparing frequency lists from different time periods. The Appendix shows keywords calculated from the 2021 *French Araneum Francogalicum* with the previous version of the same corpus used as a reference. Most items indicate the pandemic discourse prevailing in the respective time period. The presence of English words is because sentence-level language filtration has not been performed yet.

## 10 Conclusions, New Challenges and Further Work

The main *Aranea* corpus portal[25] currently hosts corpora for 27 languages, with four of them also having territorial varieties (based on the top-level domain). The "sandbox" portal (accessible for all registered users) hosts the beta versions of corpora for 7 additional languages. New crawling sessions for all languages performed since 2021 yielded data covering the discourse of the pandemic and the Russian-Ukrainian war conflict. The new crawls yielded huge amounts of data that are still being processed.

Newer versions of *Aranea* corpora are lemmatized and PoS tagged using the ensemble approach, which can significantly improve the quality of annotation. Several other improvements, including sentence-level language filters, have been incorporated into the processing pipeline. We hope to be able to reprocess all languages using this new pipeline in the near future.

New challenges include the option of using datasets from the Common Crawl[26] [18] initiative provided by the OSCAR Project[27] [1]. An important point for deliberation is whether the increasing amount of AI-generated texts means that the

[25] http://aranea.juls.savba.sk/guest/,http://unesco.uniba.sk/guest/
[26] https://commoncrawl.org/
[27] https://oscar-project.org/

web-crawled corpora will lose their importance as natural language resources in the future.

**Acknowledgments.** This work has been, in part, supported by the Slovak VEGA Grant Agency for Science, Projects No. 2/0015/14 (2014–2016), 2/0017/17 (2017–2020), and 2/0016/21 (2021–2024), respectively.

**Disclosure of Interests.** The author has no competing interests to declare relevant to this article's content.

## References

1. Abadji, J., Suarez, P.O., Romary, L., Sagot, B.: Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. In: Thirteenth Language Resources and Evaluation Conference-LREC 2022 (2022)
2. Baroni, M., Bernardini, S.: BootCaT: Bootstrapping Corpora and Terms from the Web. In: LREC. pp. 1313–1316 (2004)
3. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language resources and evaluation **43**, 209–226 (2009)
4. Benko, V.: Data Deduplication in Slovak Corpora. In: Slovko 2013: Natural Language Processing, Corpus Linguistics, E-learning. pp. 27–39. RAM-Verlag: Lüdenscheid (2013)
5. Benko, V.: Aranea: Yet Another Family of (Comparable) Web Corpora. In: Text, Speech and Dialogue: 17th International Conference, TSD 2014, Brno, Czech Republic, September 8-12, 2014. Proceedings 17. pp. 247–256. Springer (2014)
6. Benko, V.: Aranea Go Middle East: Persicum. In: RASLAN. pp. 113–121 (2022)
7. Benko, V., Kunilovskaya, M.: Comparable Web-Crawled Corpora as a Resource for Contrastive Studies. In: Accepted for presentation at the Between Languages: Methods of Contrastive Research Using Corpora Workshop, Biennial of Czech Linguistics (2024)
8. Benko, V., Zakharov, V.: Very Large Russian Corpora: New Opportunities and New Challenges. In: Kompjuternaja lingvistika i intellektuaľnyje technologii. pp. 83–98 (2016)
9. Davies, M.: The best of both worlds: Multi-billion word "dynamic" corpora. In: Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7). pp. 23–28 (2019)
10. Halácsy, P., Kornai, A., Oravecz, C.: HunPos – An open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. pp. 209–212 (2007)
11. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family. Corpus Linguistics 2013 p. 125 (2013)
12. Jongejan, B., Dalianis, H.: Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 145–153 (2009)

13. Khanna, T., Washington, J.N., Tyers, F.M., Bayatlı, S., Swanson, D.G., Pirinen, T.A., Tang, I., Alos i Font, H.: Recent advances in Apertium, a free/open-source rule-based machine translation platform for low-resource languages. Machine Translation **35**(4), 475–502 (2021)
14. Ljubešić, N., Klubička, F.: {bs,hr,sr}WaC – Web corpora of Bosnian, Croatian and Serbian. In: Proceedings of the 9th web as corpus workshop (WaC-9). pp. 29–35 (2014)
15. McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., et al.: Universal dependency annotation for multilingual parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 92–97 (2013)
16. Michelfeit, J., Pomikálek, J., Suchomel, V.: Text Tokenisation Using unitok. In: RASLAN. pp. 71–75 (2014)
17. Paikens, P.: Deep Neural Learning Approaches for Latvian Morphological Tagging. In: Human Language Technologies–The Baltic Perspective, pp. 160–166. IOS Press (2016)
18. Patel, J.M., Patel, J.M.: Introduction to common crawl datasets. Getting structured data from the internet: running web crawlers/scrapers on a big data production scale pp. 277–324 (2020)
19. Pomikálek, J.: Removing Boilerplate and Duplicate Content from Web Corpora. Ph.D. thesis, Masaryk University, Faculty of Informatics, Brno, Czech Republic (2011)
20. Rabin, M.O.: Fingerprinting by random polynomials. Technical report (1981)
21. Rychlý, P.: Manatee/Bonito – A Modular Corpus Manager. In: Recent Advances in Slavonic Natural Language Processing (RASLAN 2007). pp. 65–70 (2007)
22. Schäfer, R., Bildhauer, F., et al.: Building large corpora from the web using a new efficient tool chain. In: Lrec. pp. 486–493 (2012)
23. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: New methods in language processing. pp. 154–164. Routledge (2013)
24. Spoustová, D., Hajič, J., Raab, J., Spousta, M.: Semi-Supervised Training for the Averaged Perceptron POS Tagger. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). pp. 763–771 (2009)
25. Straka, M., Hajič, J., Straková, J.: UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 4290–4297 (2016)
26. Suchomel, V.: Discriminating Between Similar Languages Using Large Web Corpora. In: Recent Advances in Slavonic Natural Language Processing (RASLAN 2019). p. 129 (2019)
27. Suchomel, V., Pomikálek, J., et al.: Efficient web crawling for large text corpora. In: Proceedings of the seventh Web as Corpus Workshop (WAC7). pp. 39–43 (2012)

# Appendix

| | *Araneum Francogallicum Novum MMXXI Duplex Minus (Global French French, 21.03) 125 M* | | *Araneum Francogallicum III Minus (Global French, 20.05) 125 M* | | |
|---|---|---|---|---|---|
| **lemma** | **document frequency** | **document frequency/mill** | **document frequency** | **document frequency/mill** | **Score** |
| **Covid-19** | 2,576 | 20.6 | 0 | 0.0 | 21.6 |
| **coronavirus** | 1,892 | 15.1 | 4 | 0.0 | 15.6 |
| **confinement** | 2,893 | 23.1 | 129 | 1.0 | 11.9 |
| **pandémie** | 2,768 | 22.1 | 126 | 1.0 | 11.5 |
| **COVID-19** | 1,287 | 10.3 | 0 | 0.0 | 11.3 |
| **Covid** | 962 | 7.7 | 0 | 0.0 | 8.7 |
| **Coronavirus** | 563 | 4.5 | 0 | 0.0 | 5.5 |
| **déconfinement** | 372 | 3.0 | 1 | 0.0 | 3.9 |
| **covid** | 354 | 2.8 | 0 | 0.0 | 3.8 |
| **COVID** | 345 | 2.8 | 0 | 0.0 | 3.8 |
| **télétravail** | 627 | 5.0 | 85 | 0.7 | 3.6 |
| **covid-19** | 295 | 2.4 | 0 | 0.0 | 3.4 |
| **distanciation** | 524 | 4.2 | 79 | 0.6 | 3.2 |
| **Biden** | 286 | 2.3 | 20 | 0.2 | 2.8 |
| **couvre-feu** | 437 | 3.5 | 74 | 0.6 | 2.8 |
| **épidémie** | 2,188 | 17.5 | 695 | 5.6 | 2.8 |
| **that** | 568 | 4.5 | 140 | 1.1 | 2.6 |
| **which** | 282 | 2.3 | 33 | 0.3 | 2.6 |
| **cookies** | 1,094 | 8.8 | 357 | 2.9 | 2.5 |
| **RGPD** | 281 | 2.2 | 39 | 0.3 | 2.5 |
| **visioconférence** | 400 | 3.2 | 89 | 0.7 | 2.5 |
| **Castex** | 219 | 1.8 | 17 | 0.1 | 2.4 |
| **LREM** | 301 | 2.4 | 51 | 0.4 | 2.4 |
| **sanitaire** | 5,768 | 46.1 | 2,373 | 19.0 | 2.4 |
| **autrice** | 213 | 1.7 | 23 | 0.2 | 2.3 |
| **présentiel** | 385 | 3.1 | 99 | 0.8 | 2.3 |
| **this** | 505 | 4.0 | 152 | 1.2 | 2.3 |
| **reconfinement** | 158 | 1.3 | 0 | 0.0 | 2.3 |
| **their** | 243 | 1.9 | 40 | 0.3 | 2.2 |
| **also** | 219 | 1.8 | 30 | 0.2 | 2.2 |
| **Macron** | 1,847 | 14.8 | 770 | 6.2 | 2.2 |
| **based** | 216 | 1.7 | 32 | 0.3 | 2.2 |
| **has** | 328 | 2.6 | 84 | 0.7 | 2.2 |
| **hydroalcoolique** | 159 | 1.3 | 9 | 0.1 | 2.1 |
| **Netflix** | 560 | 4.5 | 202 | 1.6 | 2.1 |