M A T E M A T I C K O - F Y Z I K Á L N Í F A K U L T A

P R A H A

UW2C – LARGE MULTILINGUAL CORPUS

MARTIN MAJLIŠ, ZDENĚK ŽABOKRTSKÝ

ÚFAL/CKL Technical Report
TR-2011-46

U N I V E R S I T A S C A R O L I N A P R A G E N S I S

Copies of ÚFAL/CKL Technical Reports can be ordered from:

Institute of Formal and Applied Linguistics (ÚFAL MFF UK)
Faculty of Mathematics and Physics, Charles University
Malostranské nám. 25, CZ-11800 Prague 1
Czech Republic

or can be obtained via the Web: http://ufal.mff.cuni.cz/techrep

Charles University in Prague

Faculty of Mathematics and Physics

Technical Report

Martin Majliš

Zdeněk Žabokrtský

W2C - Large Multilingual Corpus

Institute of Formal and Applied Linguistics

Prague 2011

Contents

1. Introduction

As statistical approaches become the dominant paradigm in natural language processing, there is an increasing demand for data. It is known that simple models and a lot of data outclass sophisticated models based on less data. The web contains huge amounts of linguistics data for many languages. The web has many undeniable advantages: (a) size — it is the largest text collection containing billions of documents and its size is exponentially growing, (b) range — texts are available in many languages, styles and domains, (c) availability — most of the documents are available in machine-readable form, so no scanning or rewriting is necessary.

One of the key issues for computational linguists is easy access to such data, but already collected corpora are available only for the major word languages.

Therefore, our aim is to collect, with minimal or no human intervention, at least ten millions of words for as many languages as possible.

1.1 Problem Definition

The goal of this project is to build multilingual corpus of texts available on the Internet. This corpus will consist of as many words as possible for as many languages as possible. The collected material will be quantitative and qualitative analysed.

The project consists of:

- A study of existing multilingual resources and approaches used to construct them.
- A review of tools and methods used for solving particular tasks such as building initial corpus, crawling, language recognition, and duplicity detection.
- A design for solving these particular tasks as well as the main tasks with respect to amount of processed data.
- An implementation of tools and processes capable of taking benefits of distributed environment.
- A quantitative and qualitative analyses of the collected material.

- Conclusions about used methods with evaluation of their performances for different languages.

1.2 Motivation

There are many publicly available projects that are trying to collect multilingual textual resources. Some of them cover many languages, but contain either very few documents or these documents are not in computer accessible form, so they cannot be easily used in computational linguistics. Other projects contain more data, but are available in very few languages. Therefore, it will be useful to construct corpus, that will overcome these disadvantages. When this data becomes available, it will be possible to use it for comparative analysis of related languages, building language models for various applications such as machine translation, speech recognition, spell checking, etc. For achieving the main goal, many subtasks has to be solved, such as identifying languages or downloading millions of web pages. When all this data is collected, it will be possible to use it for further improvements.

1.3 Report Organization

The work is divided into five chapters, beginning with the introductory Chapter 1 containing problem definition and motivation. Overview of existing methods and techniques is presented in Chapter 2. This chapter briefly introduces existing multilingual resources and multilingual corpora as well as methods used for their construction. It also presents methods for solving particular steps. Requirements for the complete system and available computational resources are described in Chapter 3. It also introduces implemented tools and methods for their effective usage. Achieved results in language identification and size of constructed corpus are shown in Chapter 4. A quantitative and qualitative analyses of the corpus is included. Overall results are discussed in Chapter 5 as well as areas where the methods and implementation could be improved. It also suggests goals for the for future work.

Two appendices are included: lists of languages covered by the collected corpus with their ISO-639-3 codes is presented in Appendix A. Differences between

languages included in the W2C Wiki Corpus and the W2C Web Corpus are presented in Appendix B.

2. Literature Review

This chapter reviews existing tools, methods, and approaches. It opens by presenting statistics about existing languages, followed by an introduction of existing multilingual projects. The main part of this chapter is an overview of multilingual web corpora as well as methods used for crawling, text extraction, language identification, corpus storing, and distribution.

There are 6,909 known living languages according to the Ethnologue database,[1] but only about 390 of them are used by more than 1 million of native speakers,[2] while 172 of them have more than 3 million speakers.

2.1 Language Resources

There are many projects aim to collect materials in as many languages as possible, because there are predictions that fifty percent of the world's languages will disappear in the next century.[3]

Detailed distribution of languages and speakers is showed in Table 2.1. These numbers must be treated with caution, because they are slightly out-of-date. Total population according to this table is 6 billion, but it was true in 1999.[4]

Following projects are reviewed:

- The Rosetta Project (2.1.1)
- The Open Language Archives Community (2.1.2)
- The Wikipedia (2.1.3)
- The Universal Declaration of Human Rights (2.1.4)
- The Project Gutenberg (2.1.5)
- The Wikisource (2.1.6)
- The Watchtower (2.1.7)
- Open-source Software (2.1.8)

[1] http://www.ethnologue.com/web.asp
[2] http://www.ethnologue.com/ethno_docs/distribution.asp?by=size
[3] http://www.unesco.org/new/en/culture/themes/cultural-diversity/languages-and-multilin
[4] http://www.census.gov/population/international/data/idb/worldpopgraph.php

Population range	Living languages			Number of speakers		
	Count	Percent	Cumulative	Count	Percent	Cumulative
100,000,000 to infinity	8	0.1	0.1%	2,308,548,848	38.73721	38.73721%
10,000,000 to 99,999,999	77	1.1	1.2%	2,346,900,757	39.38076	78.11797%
1,000,000 to 9,999,999	304	4.4	5.6%	951,916,458	15.97306	94.09103%
100,000 to 999,999	895	13.0	18.6%	283,116,716	4.75067	98.84170%
10,000 to 99,999	1,824	26.4	45.0%	60,780,797	1.01990	99.86160%
1,000 to 9,999	2,014	29.2	74.1%	7,773,810	0.13044	99.99204%
100 to 999	1,038	15.0	89.2%	461,250	0.00774	99.99978%
10 to 99	339	4.9	94.1%	12,560	0.00021	99.99999%
1 to 9	133	1.9	96.0%	521	0.00001	100.00000%
Unknown	277	4.0	100.0%			
Total	6,909	100.0		5,959,511,717	100.00000	

Table 2.1: Distribution of languages by number of first-language speakers

2.1.1 Rosetta Project

The Rosetta[5] Project is a global collaboration of language specialists and native speakers working on a publicly accessible digital library of material on all known human languages. The collection currently contains nearly 100,000 pages of material spanning over 2,500 languages as well as a growing multimedia collection of modern and historical language recordings.

2.1.2 Open Language Archives Community

The Open Language Archives Community[6] (OLAC) is an international partnership of institutions and individuals who are creating a worldwide virtual library of language resources. Their language coverage is presented in Table 2.2.

2.1.3 Wikipedia

Wikipedia[7] is a free, web-based, collaborative, and multilingual encyclopedia project. It contains 19 million articles in 281 languages.[8] Article counts are presented in Table 2.3.

[5] `http://rosettaproject.org/` and `http://www.archive.org/details/rosettaproject`
[6] `http://www.language-archives.org/`
[7] `http://www.wikipedia.org/`
[8] `http://meta.wikimedia.org/wiki/List_of_Wikipedias`

Population range	Languages	Coverage			Online Resources		
		Count	Percent	Items	Count	Percent	Items
100,000,000 to 999,999,999	8	8	100%	7745	8	100%	1007
10,000,000 to 99,999,999	77	75	97%	4367	72	94%	2152
1,000,000 to 9,999,999	304	277	91%	4887	246	81%	3006
100,000 to 999,999	895	716	80%	8814	600	67%	4388
10,000 to 99,999	1824	1181	65%	15208	951	52%	5581
1,000 to 9,999	2014	1244	62%	20566	1097	54%	8190
100 to 999	1038	634	61%	11239	560	54%	3799
10 to 99	339	235	69%	6427	202	60%	1075
1 to 9	133	90	68%	1067	75	56%	519
Unknown	277	115	42%	1731	79	29%	394
All living languages	6909	4575	66%	82051	3890	56%	30111
Extinct languages	520	242	47%	2328	178	34%	778

Table 2.2: OLAC – language coverage

Articles	Count	Cumulative
1,000,000 to 9,999,999	3	3
100,000 to 999,999	34	37
10,000 to 99,999	64	101
1,000 to 9,999	107	208
100 to 999	60	268
10 to 99	7	275
1 to 9	5	280

Table 2.3: Wikipedia – article counts

2.1.4 Universal Declaration of Human Rights

The Universal Declaration of Human Rights[9] (UDHR) is a milestone document in the history of human rights. At present, there are 379 different translations of UDHR, available in HTML and/or PDF format. There is a related project UDHR in Unicode[10] which aims to convert all documents into Unicode.

2.1.5 Project Gutenberg

The Project Gutenberg[11] is a volunteer effort to digitize and archive cultural works. It contains over 34 thousands documents in 60 languages and most of them are texts of public domain books.

2.1.6 Wikisource

Wikisource[12] is an online library of free content textual sources, operated by the Wikimedia Foundation. Its aims are to harbour all forms of free text, in many languages. Wikisource contains more than one million articles in 62 languages.[13]

2.1.7 Watchtower

The Watchtower[14] is an illustrated religious magazine, published semi-monthly by Jehovah's Witnesses. It is written in 418 languages (366 without sign languages). Texts are available as web pages or PDF files. All files have a very similar structure, so it may serve as a very good source of parallel texts.

2.1.8 Open-source Software

Open-source Software[15] (OSS) is computer software that is available in source code typically developed by volunteers distributed amongst different geographic

[9] http://www.ohchr.org/EN/UDHR/Pages/Introduction.aspx
[10] http://unicode.org/udhr/
[11] http://www.gutenberg.org/
[12] http://www.wikisource.org/
[13] http://meta.wikimedia.org/wiki/Wikisource#List_of_Wikisources
[14] http://watchtower.org/
[15] http://en.wikipedia.org/wiki/Open_source_software

regions. Therefore, big OSS projects are available in many languages. Texts that can be extracted are mostly error messages, menus, and buttons. There are for example the following OSS projects:

- Launchpad[16] — 323 languages, 1,730,838 strings
- Gnome[17] — 173 languages
- KDE[18] — 75 languages

2.1.9 Summary

Sizes of different language resources are summarized in Table 2.4. From these sizes, it is possible to conclude:

- Thousands of languages are available in the Rosetta Project and the Open Language Archives Community. Special language interest groups and linguistics specialists are required to achieve this amount of languages.
- Around 300 languages are presented in the Universal Declaration of Human Rights, Wikipedia, the Watchtower, and Launchpad. This is the upper bound for number of languages that are at least theoretically available in written form on the Internet. This covers almost 90% of all people.
- Around 60 languages are available in Project Gutenberg and Wikisource. This is the lower bound for the number of languages that are used in developed or newly industrialized countries[19] countries. This covers almost 70% of all people.

2.2 Multilingual Web Corpora

As early as 2001, Banko and Brill [BB01] and recently in 2009 Halevy et al. [HNP09] showed that using more data and simple method outperform less data and sophisticated method.

Therefore many scientists were collecting multilingual resources. The good source of multilingual texts is the Internet and especially web pages. For that reason

[16] `https://translations.launchpad.net`
[17] `http://l10n.gnome.org/languages/`
[18] `http://l10n.kde.org/teams-list.php`
[19] `http://en.wikipedia.org/wiki/Newly_industrialized_country`

Projects	Languages	Size
Rosetta Project 2.1.1	over 2,500	100,000 pages
OLAC 2.1.2	4,575	82,051 items
Wikipedia 2.1.3	281	19,034,746 articles
UDHR 2.1.4	379	at most 379 documents
Project Gutenberg 2.1.5	60	34,000 documents
Wikisource 2.1.6	62	1,028,303 pages
Watchtower 2.1.7	366	thousands of pages
Launchpad 2.1.8	323	1,730,838 strings
Gnome 2.1.8	173	about 1 million of strings

Table 2.4: Multilingual resources — summary

there are many already existing multilingual web corpora, that are using almost unified approaches for their construction. In this section following projects are reviewed in more details:

- WaCky (2.2.1)
- Crúbadán (2.2.2)
- I-X (2.2.3)
- Corpus Factory (2.2.4)

The unit 'W' will be used instead of word, so 10 MW means 10 million words.

2.2.1 WaCky

WaCky was introduced for the first time by Baroni and Kilgarriff [BK06] in 2006 with more detailed information in [BBFZ09]. This corpus contains 3 languages - English, German and Italian — and each of them has approximately 1.5 TW.

Building a corpus for each language took approximately 3 weeks (10 days crawling, 7 days cleaning, 4 days near-duplicate detection). Basic statistics are presented in Table 2.5.

2.2.2 Crúbadán

Crúbadán is a multilingual corpus introduced by Scannel [Sca07]. This corpus contains 487 languages.[20] Crúbadán corpus size is presented in Table 2.6.

[20] `http://borel.slu.edu/crubadan/stadas.html`

Property	deWaC	itWaC	ukWac
Raw crawl size (GB)	398	379	351
Documents after filtering (M)	4.86	4.43	5.69
Size after document filtering (GB)	20	19	19
Size after near-duplicate cleaning (GB)	13	10	12
Documents after near-duplicate cleaning (M)	1.75	1.87	2.69
Tokens (G)	1,278	1,586	1,914

Table 2.5: WaCky — data size

(a) Document counts

Document count	Languages
> 1000	70
> 500	115
> 250	143
> 125	181
> 65	210
> 32	255
> 16	337
> 8	356
> 4	381
> 2	416
> 1	449

(b) Word counts

Word count	Languages
> 100 MW	1
> 10 MW	11
> 1 MW	127
> 100 kW	225
> 10 kW	354
> 1 kW	473
> 100 W	487

Table 2.6: Crúbadán — data size

Language	Size in MW
English (I-EN)	127
German (I-DE)	126
Russian (I-RU)	156
Chinese	???
Romanian	???
Ukrainian	???

Table 2.7: I-X — size in MW

Language	Wiki Corpus	Web Corpus
Dutch	30.0	108.6
Hindi	2.5	30.6
Indonesian	8.5	102.0
Norwegian	19.1	94.9
Swedish	9.3	114.0
Telugu	0.2	3.4
Thai	6.2	81.8
Vietnamese	9.5	149.0

Table 2.8: Corpus Factory — size in MW

2.2.3 I-X

Sharoff [Sha06] introduced BNC-like multilingual web corpus. This corpus[21] contains 6 languages — English, German, Russian, Chinese, Romanian, and Ukrainian, but only for three of them are results available.

The corpus size is presented in Table 2.7. The corpora for Chinese, Romanian, and Ukrainian are mentioned only in the introduction and no results for them are presented.

2.2.4 Corpus Factory

Corpus Factory is a multilingual corpus constructed by Kilgarriff [KRPP10]. This corpus contains 8 languages - Dutch, Hindi, Indonesian, Norwegian, Swedish, Telugu, Thai, and Vietnamese. Corpus size is displayed in Table 2.8.

[21] `http://corpus.leeds.ac.uk/internet.html`

2.2.5 Summary

In this subsection we summarize existing multilingual corpora and compare them with one another. Sizes are presented in Table 2.10. All approaches used very similar methods:

1. Retrieve word seeds from existing corpus or reliable text source.
2. Generate n-tuples of words.
3. Use these tuples as search queries.
4. Download found web pages.
5. Preserve just files with mime text/html and acceptable size.
6. Remove boilerplate code.
7. Use functional words for language detection and running text detection.
8. Use Broder's "shingling" algorithm ([BGMZ97]) to find near duplicate detection.

Differences among all approaches are displayed in Table 2.9.

2.3 Corpus Storing and Distribution

Corpus storing and distribution is one of the fundamental parts of corpus building. Wynne ([Wyn05]) as well as E-MELD[22] suggests many tips.

Archival copies should be made in a format which offers LOTS (i.e., it is Lossless, Open Standard, Transparent, and Supported by multiple vendors). A corpus must also contain proper documentation of used formats along with information about terms of use and access restrictions.

Making a corpus widely available should not be possible due to copyright and other legal issues.

2.4 Corpus Quality Analysis

Corpus quality analysis is also an important step in building web corpus. Without comparing with existing corpora it is hard to say whether high quality texts were downloaded or if they are just some 'CD image'.

[22] `http://emeld.org/school/bpnutshell.html`

Property	WaCky	Crúbadán	I-X	Corpus Factory
Word seeds	Texts from existing corpora.	Texts from from specified website.	Texts from existing corpora.	Texts from Wikipedia.
URL seeds	Searching pairs of mid-frequency content words using google.	Searching randomly chosen words from lexicon (OR'ed together) with AND'ed at least one stopword.	Searching randomly chosen words from lexicon (AND'ed together) with OR'ed 2 high frequency words.	Searching mid-frequency words. Number of words is language dependent.
Crawler	Heritrix	wget	Unspecified	wget
Crawling	Domain restricted, suffix restricted. Recursive.	Extracted URLs are added to the pending list of URLs for the language of the downloaded document. Recursive.	Just extracted URLs. Without recursion.	Just extracted URLs. Without recursion.
Filtering	Mime type text/html, size between 5 kB and 200 kB.	Unmentioned	Unmentioned	Mime type text/html, size between 5 kB and 2 MB. At least 65% of high frequency words.
Boilerplate	Modified BTE algorithm.	Unmentioned	Tag density (maybe BTE)	BTE algorithm.
Deduplication	Simplified version of Broder's "shingling" algorithm.	Unspecified	Simplified version of Broder's "shingling" algorithm.	Broder's "shingling" algorithm.
Language Detection	Contains functional words.	Cosine angle between vectors representing the document and training texts in the space of character trigrams. Manual tuning.	Unmentioned. Functional words in search query.	Unmentioned. Functional words in search query.
Languages	3	487	3 (6)	8
Median size	1.586 GW	68,221 W	126 MW	102 MW

Table 2.9: Existing multilingual corpora — overview

Language	WaCky	Crúbadán	I-X	Corpus Factory
English	1,914 GW	26.8 MW	127 MW	No
German	1,278 GW	2.7 MW	126 MW	No
Russian	No	333 kW	156 MW	No
Italian	1,586 GW	3.2 MW	No	No
Dutch	No	2.6 MW	No	138.6 MW
Hindi	No	805 kW	No	33.1 MW
Indonesian	No	5 MW	No	110.5 MW
Norwegian	No	2.6 MW (N)	No	114 MW
Swedish	No	2 MW	No	123.3 MW
Telugu	No	2 MW	No	3.6 MW
Thai	No	218 kW	No	90 MW
Vietnamese	No	3.9 MW	No	158.5 MW
Chinese	No	320 kW	Yes	No
Romanian	No	6.6 MW	Yes	No
Ukrainian	No	273 kW	Yes	No

Table 2.10: Language coverage

Rayson et. al [RG00] suggested using log-likelihood statistics for comparing frequency lists. Bharati et. al [BRSB00] also suggested using a number of unique unigrams, entropy, word and sentence lengths for comparing different corpora.

3. Methods

This chapter describes tools and methods used for building web corpus. Complete process is illustrated on Figure 3.1 with available resources and data flow.

Constructing of the web corpus consists of several steps. The initial step was gathering metadata from Wikipedia and Ethnologue. The downloaded metadata was stored in the database on the hosting. When matadata was available, then a wiki corpus was built from Wikipedia articles. Frequency lists for trigrams and quadgrams were computed and uploaded to the hosting. From the wiki corpus the language model was trained and moved to the hosting. Building a web corpus was divided into smaller jobs that were executed in the computer laboratory. Job results were stored on ufallab where they were merged into raw corpus. This raw corpus was transferred back to the UFAL cluster where the downloaded pages were reprocessed with improved language identifier. From this data duplicities were removed, statistics were computed, and packages for distribution were prepared.

3.1 Metadata

Metadata, such as language name, its ISO code, population size, writing system, etc., was for each language automatically downloaded from the Internet. The following sources were combined:

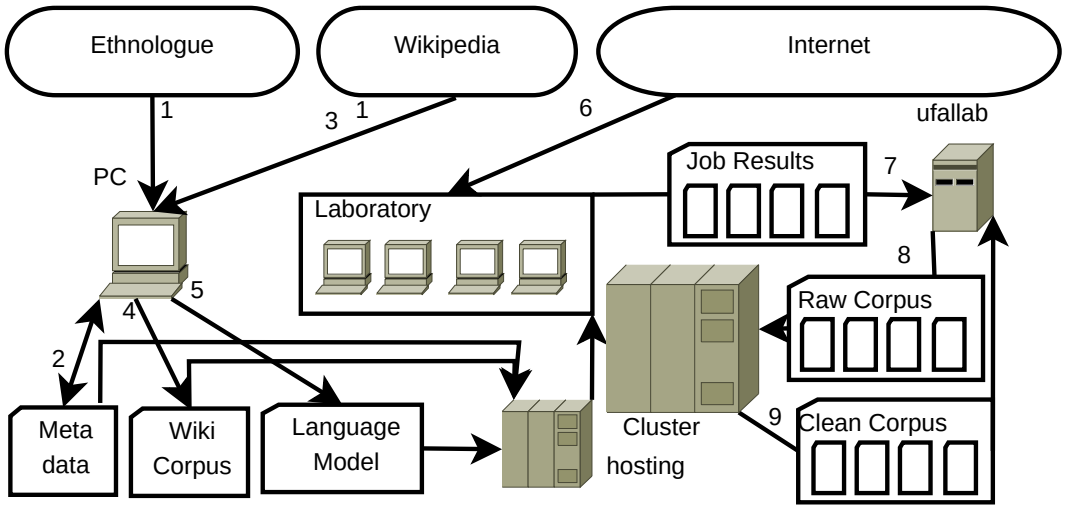

Figure 3.1: Building Web Corpus

- SIL International[23] — which provides easily parsable table[24] of all languages with their ISO codes and names.
- Wikipedia[25] — with its list of all wikipedias[26] where they use their own codes and names.
- Ethnologue[27] — with easily parsable pages with language information - e.g. Czech.[28]

Because we knew that the Ethnologue numbers are out-of-date (2.1), we intended to use information from the info-boxes in Wikipedia. For example, English has 328 million speakers according to Ethnologue,[29] while Wikipedia[30] provides also information about first and second language speakers with overall up to 1.8 billion speakers. In fact, English is the 'Lingua franca' of the Internet therefore we would prefer to use numbers from Wikipedia.

To avoid parsing Wikipedia, we wanted to use DBpedia,[31] which extracts information from Wikipedia, but we discovered that it is not reliable. For example, for the Buginese language DBPedia:[32] 240 speakers, Wikipedia:[33] 3.5 to 4 millions and Ethnologue: [34] 3.5 millions.

From this we concluded that information extraction from Wikipedia may not be suitable. Not all languages are present on Wikipedia and it may be hard to localize them, due to their name variants. It would be also hard to automatically and correctly decide which number of speakers is correct. Therefore, we decided to stick with Ethnologue.

Scripts used for metadata extraction are `langList.sh` and `ethnologueParser.sh`.

In the early stages, extracted information was stored in text files. Later on, they was moved to the database.

[23] `http://sil.org`
[24] `http://www.sil.org/iso639-3/iso-639-3_20100707.tab`
[25] `http://www.wikipedia.org/`
[26] `http://meta.wikimedia.org/wiki/List_of_Wikipedias`
[27] `http://www.ethnologue.com/`
[28] `http://www.ethnologue.com/show_language.asp?code=ces`
[29] `http://www.ethnologue.com/show_language.asp?code=eng`
[30] `http://en.wikipedia.org/wiki/English_language`
[31] `http://dbpedia.org/`
[32] `http://dbpedia.org/page/Buginese_language`
[33] `http://en.wikipedia.org/wiki/Buginese_language`
[34] `http://www.ethnologue.com/show_language.asp?code=bug`

3.1.1 Access

There are three ways how to access stored data - using web interface, simplified RESTful API,[35] and script `webAPI.sh`.

The web interface is available on `http://w2c.martin.majlis.cz/language/`. It is possible to specify the language and key and all corresponding values are returned. It is possible to specify output format which can be:

- TXT — text output – columns are separated by tabs. This output may be easily processed with unix command-line tools.
- XML — XML output
- JSON — JSON[36] output which can be easily used in programs.

The URLs provided by the web interface are also a part of the REST API. If proper authentication token is used, values may be changed or new ones added.

The script `webAPI.sh` is a wrapper written in bash. It uses REST API and its text output. This script is used by almost all programs.

3.1.2 Work Flow

Metadata is automatically retrieved from the Internet with scripts `langList.sh` and `ethnologueParser.sh`. Downloaded information is stored in temporary text files. These files are then processed with scripts in a `fillLangDB` directory. These scripts use `webAPI.sh` for inserting data into the database. When any script (`S1`, `S2` etc.) needs any information, it uses `webAPI.sh`. Some scripts are also adding new metadata, therefore an arrow exists between scripts and `webAPI.sh` is bidirectional. This worfkflow is depicted in in Figure 3.2.

Using this metadata, it is very easy to create simple scripts. In Example 1 is shown simple script that creates corpus from Wikipedia articles in all languages that are not using Latin script.

[35] `http://en.wikipedia.org/wiki/REST`
[36] `http://en.wikipedia.org/wiki/JSON`

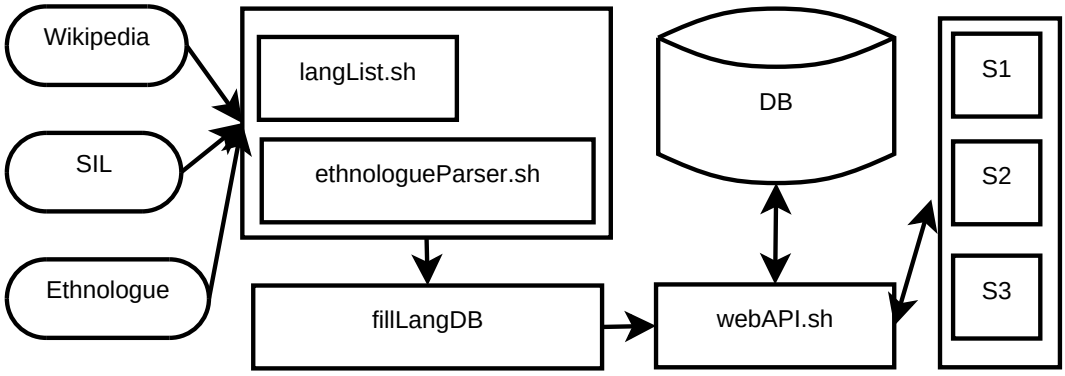

Figure 3.2: Metadata — work flow

Example 1 Building wiki corpora for languages not using latin script

```
for l in 'webAPI.sh GET null script | grep -v 'Lat' | cut -f1'; do
        url='webAPI.sh GET $l 'wiki url' | cut -f3';
        if [ ! -z $url ]; then
                wikiCorpus.sh -c 100 $l;
        fi;
done;
```

3.2 W2C Wiki Corpus

The next step in building a web corpus was to construct the initial corpus. We decided to use Wikipedia (2.1.3), because it was widely used in other multilingual corpora and also, we have previously worked with Wikipedia. We constructed several tools, developed a work flow for building wiki corpus, and built wiki corpus containing hundred languages.

3.2.1 Tools

Script `wikiCorpus.sh` downloads directly the Wikipedia dumps (provided by Wikimedia). We used the CPAN module Text::MediawikiFormat[37] to convert the wiki format to HTML and then to plain text. We found out that this module did not work correctly, so we used slightly different approach. At the beginning all links, tables and special syntax were removed. This preprocessed text was passed to the Text::MediawikiFormat module to create a HTML output, from which only paragraphs were preserved and all tags are removed. In the last phase duplicate lines were removed with the script `cleanFile.sh`.

[37] `http://search.cpan.org/~dprice/Text-MediawikiFormat-0.05/`

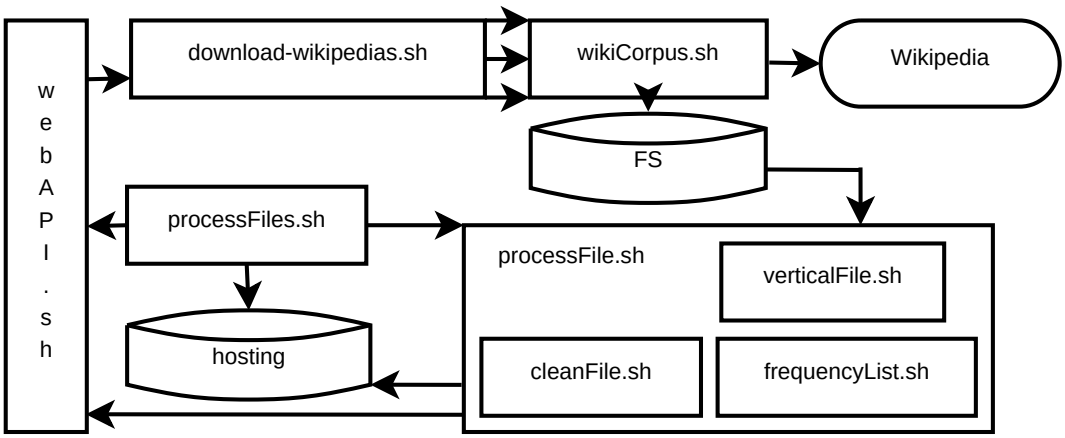

Figure 3.3: W2C Wiki Corpus — work flow

3.2.2 Data

We used a corpus build from 5,500 articles for each language with at least 100 thousand articles for prototyping. Later on, we extended this corpus to languages with at least 5 thousands articles. This corpus contained 115 languages.

For our main work, we used a corpus of 20,000 articles from Wikipedias with at least 5 thousands articles. This corpus has a database key `data wiki_20000`.[38]

3.2.3 Work Flow

The work flow for building the W2C Wiki Corpus is displayed in Figure 3.3. The first step was script `download-wikipedias.sh` execution with a specified number of required pages and minimal article counts. This script executes `wikiCorpus.sh` for each language. Script `wikiCorpus.sh` downloaded and extracted texts from Wikipedia which were stored on the disk.

It is possible to extend this process by executing script `processFiles.sh`, which iterates over languages downloaded in the first step. For each language script `processFile.sh` was executed, which removes duplicity with `cleanFile.sh` and generates a vertical file using `verticalFile.sh`. Frequency lists for n-grams were constructed with `frequencyList.sh`. All created files were uploaded to the hosting and URLs of these files were added to the database.

[38] `http://w2c.martin.majlis.cz/language/?lang=&key=data+wiki_20000*&format=TXT`

Ratio	1-gram	2-gram	3-gram	4-gram	5-gram
0.05	0.021	0.403	0.891	0.992	0.999
0.10	0.022	0.623	0.969	0.999	0.999
0.15	0.037	0.790	0.989	0.999	0.999
0.20	0.117	0.880	0.992	0.999	0.999
0.25	0.222	0.918	0.992	0.999	0.999
0.30	0.285	0.907	0.993	0.999	0.999
0.35	0.350	0.930	0.993	0.999	0.999
0.40	0.219	0.903	0.993	0.999	0.999

Table 3.1: Language identification for the first 31 languages

3.3 Language Identification

The language identification is one of the crucial components of the project. Existing solutions, described in Section 2.2, are usually able to identify around 10 languages. To achieve the goal, our language identifier must be capable of identifying more than ten times more languages.

3.3.1 Prototype

We started language identification with simple prototyping. We built a Wikipedia corpus for languages with at least 100 thousand articles (31 at that time) and we used two thousand of them. We used the simplest method - character n-gram model. We trained it on full sentences without segmentation or any preprocessing. For example 'I am' would create 3-grams: '__I', '_I ', 'I a', ' am', 'am_' and 'm__'. We trained this model for n-grams for n from 1 to 5 and we selected n-grams from the top of the frequency list until p percent of the total n-gram count was chosen. This means that for frequency list of unigrams: 'a': 5, 'b' 2, 'c': 1 and p equals 0.5, only 'a' would be chosen. Achieved results are shown in Table 3.1. It seemed that anything more than 4-grams would provide sufficient results and we considered this problem as solved.

3.3.2 Full Scale

In the next step, we ran this experiment in full scale with more than one hundred languages, and we found out that accuracy dropped significantly. The reason was that for every major language, there is set of related languages. For English,

it was Welsh, Irish, Scottish Gaelic, Scots, etc. For Spanish it was Portuguese, Occitan, Catalan, Asturian, Galician, etc. For Russian, it was Bulgarian and Ukraine. The hardest was Croatian, Serbo-Croatian and Bosnian.

For example, the word 'goat' is in Occitan, Catalan, Spanish and Portuguese written as 'cabra', and in Latin, Italian and Romanian as 'capra'. Word 'bridge' is written as 'pont' in Occitan, Catalan and French, and as 'ponte' in Latin, Italian and Portuguese.

The full scale experiment used 20 thousands articles from Wikipedias with at least 5 thousand articles. One half was used for training, one third was used as heldout and the rest for evaluation. We tested various set up for parameters. For example, when the top 5% of 4-grams or more than 2000 4-grams were chosen, then all Russian texts were identified as Bulgarian (all Bulgarian was identified as Bulgarian). When we decreased the number of 4-grams to 200, only 4% of Russian texts were identified as Bulgarian (Bulgarian was still Bulgarian). When we decreased the number of 4-grams to 100, all samples were identified perfectly.

Decreasing the amount of n-grams dramatically increased the performance.

Language identification is tightly coupled with character encoding. Single language in multiple encoding can be considered as different language. So we left character n-grams and used byte n-grams. This decision has advantage, that all 4-grams has exactly 4 bytes, but on the other hand in this 4-gram can be only single character in exotic script encoded.

3.3.3 Final Version

The final version of our language identifier was constructed in the following way. The Wiki Corpus was divided into two parts. The first five sixths were used for training and the remaining data was used for evaluation. Test data for each language was divided into 500 equally large (in words) chunks. If a chunk was greater than 500 words then extra words were deleted.

Training

The probability of each 4-gram was computed using the training data and only the first 100 were preserved. These probabilities were normalized to sum up to 1.

(a) Training data

Lang	Training data
L1	bbbeaccdcdaabbbbeddc
L2	bbacceeceaedcdeabbeb

(b) Training Probabilities

Lang	a	b	c	d	e
L1	0.15	0.35	0.20	0.20	0.10
L1	0.15	0.25	0.20	0.10	0.30

(c) Language Model

Uni	Lang	Score	Uni	Lang	Score
b	L1	0.43	c	L2	0.27
b	L2	0.33	d	L1	0.29
c	L1	0.29	e	L2	0.40

(d) Detection — ‘aabbecdec‘

Lang	Computation	Score
L1	$0.00 + 0.00 + 0.43 + 0.43 + 0.00 + 0.29 + 0.29 + 0.00 + 0.29$	1.73
L1	$0.00 + 0.00 + 0.33 + 0.33 + 0.40 + 0.27 + 0.00 + 0.40 + 0.27$	**2.00**

Table 3.2: Language identification — example

Detection

During detection, the input text is preprocessed and divided into 4-grams. Probabilities retrieved during training are treated now as a score. Scores for each language are summed up and the language with the highest score is the winner.

Example

A simple example for two languages is shown in Table 3.2. In this example an unigrams language model and only the first 3 unigrams are used. Training data (a) is used to compute probabilities (b). Only the first 3 most probable unigrams for each language are preserved, normalized and stored in the language model (c). Language detection for sample input string is presented in Table (d), so the input string ‘aabbecdec‘ would be identified as L2.

3.4 URL Seeds

At the beginning we used external links from Wikipedia. These external links are stored as a SQL dumps provided by Wikimedia. For retrieving these links we used script `wikiExternalLinks.sh`. We found out that the vast majority of these links can not be used. Reasons were that the pages did not no-longer exist, were specialized websites or databases, were written in English, etc.

So we decided to use Google Search. When the user agent in the HTTP request header contained word 'bot', then Google returned HTTP Status Code 403 Forbidden. So we used user agents used by web browser.

We used trigram frequency file from the Wiki Corpora to generate search phrases. All trigrams with numbers or punctuation were removed and from the remaining list trigrams on lines from 2nd to 5th percentile were chosen. We used 30 queries to Google and stored the first hundred of links.

3.5 W2C Web Corpus

The construction of the W2C Web Corpus was divided into two separate steps. The first step involved downloading web pages from the Internet and the second step was compiling the corpus.

3.5.1 Downloading Data

The corpus was downloaded from the Internet using W2C Builder ([Maj11]). The W2C Builder is a distributed corpus builder capable of running on multiple machines and consisting of following components:

- crawler — receives an URL and returns HTML code
- parser — receives HTML code and return text
- detector — receives text and returns language code
- master — coordinates work of all components mentioned above

create-corpus.sh

For building a web corpus with 10 million words in Czech, it is sufficient to execute `./create-corpus.sh ces 10M`.

The script `create-corpus.sh` is the main script executed by the end-user. For example, the command `create-corpus.sh ces 10M` creates a corpus with 10 million words for Czech. This script is responsible for argument checking — whether specified language code is available in the language identifier. When the correct language is used, then the language model and corresponding trigram frequency list is downloaded from the hosting. The URL seed (3.4) is constructed

from the downloaded frequency lists. Then, scripts `keeper.sh` and `charter.sh` are executed in the background. Then the master `create-corpus.pl` is executed. When the master finishes `keeper.sh` and `charter.sh` are killed and the downloaded results are packed with script `packData.sh`.

create-corpus.pl

The script `create-corpus.pl` is the master script for the W2C Builder and works as a server for all workers.

During the initialization phase, the script reads the configuration file, inserts an initial URL seed into database, and builds a distribution archive. The path to the configuration file and the file with the initial URLs are passed as an arguments. The distribution archive is a gzipped tar archive with source codes necessary for worker execution. Then, distribution archives are copied on nodes specified in the configuration file and the corresponding workers are executed.

All URLs are stored in the SQLite database.[39] We decided to use this database, because it is widely available on all systems, and therefore it does not increase requirements.

Logging is important for debugging and run analysis of complex programs, so we decided to use log4perl,[40] which is compatible with log4j.[41] Apache Log4 is widely used in applications written in Java, but there are also ports for other languages. The main advantage of the widely used format is the availability of tools for log analysis.[42]

Tasks

The task is a small unit of work which is assigned to a waiting worker. The task is in the form of gzipped tar archive, designed in such a way, that the output from the preceding worker in the processing pipeline is the input for the following worker. The main file in the archive is called a protocol, columns are called attributes. Each row contains information about a processed URL.

[39] http://www.sqlite.org/
[40] http://mschilli.github.com/log4perl/
[41] http://logging.apache.org/log4j/1.2/
[42] http://en.wikipedia.org/wiki/Log4j#Log_Viewers

The crawl task contains only a protocol with URLs. URLs are read from the database. When an URL is chosen, it is marked as 'in progress'. The crawl downloads URLs and fills attributes actual time, URLs md5 hash, HTTP Status code, base URL, charset, and size. Downloaded files are added to the archive in the form of `urls-md5.html`.

The parser task is the crawler's output archive. It reads the protocol and searches for URLs with the correct attributes (HTTP status, mime-type). If a correct URL is found, the stored HTML file is processed. Links are stored in the file `urls-md5.links`, text is saved to the file `urls-md5.txt` and attributes for number of links, text size in characters and text size in words are filled in.

The detector task is the parsers's output archive. It reads the protocol and searches for URLs with the correct attributes (text size, number of links). If a correct URL is found, a language is identified and stored to the protocol.

When the server retrieves a result from any detector, it reads the protocol and searches for URLs in the target language. If a URL is found, all links are added to the database and the text is appended to the corpus. The attributes of all URLs are stored in the database and the URL itself is marked as finished.

When a new URL is added to the database, it gets assigned a random number. When URLs are selected for a new crawler task, then the first N according to this random number are chosen. This approach reduces the probability that all selected URLs will be from the same domain.

This design allows reprocessing finished tasks. If the text extraction or the language detection are improved, then all finished tasks could be used as input for the parser or detector.

URL Preprocessing and Filtering

All URLs are normalized[43] to reduce the obvious duplicity on the URL level; for example, these URLs are equal `HTTP://www.Example.com/` and `http://www.example.com/`.

The URL filtering was essential for increasing the yield of the crawling. In the early versions, we started with manually written regular expressions for the most common file types (doc, docx, xls, xlsx, etc.), which should be ignored. After a few experiments, we found out, that this is not sufficient, because lot of links directed

[43] http://en.wikipedia.org/wiki/URL_normalization

to advertisement websites. We thus decided to use a list of known advertising websites[44] as blacklist. However, further investigation revealed that there are also links to bookmarking services (digg, stumble, etc.) or social services (twitter, facebook), which should also be ignored, so we abandoned this idea.

Also, the top-level domain names can be used for filtering. When the task is to build a Czech corpus, all pages under TLD '.cz' are good candidates (Czech is used in the Czech Republic with the TLD '.cz') but pages under '.de' (Germany) are not good candidates. It would be feasible to create such rules for a few major languages, but not for hundreds. Furthermore, domains under the 'right' TLD are not always worth crawling - for example search results, catalogues, advertisement servers etc.

To solve this problem, we used an additional database with two tables - one for TLDs and one for domains. These tables contain column for the TLD (or domain name), the number of downloaded URLs, the number of valid URLs, the ratio of valid URL (in percent) and information, whether this domain is ignored.

When a URL was processed, then its TLD and domain name was extracted. The number of downloads for this TLD and domain was increased. If the URL was in the target language, than the number of valid URLs was also increased and the ratios were updated. If the TLD was downloaded more than 20 times and has less then 10% of valid URLs, then it was marked as ignored. Same approach was used for domains, but at least 40 downloads were required. The ratio 10% looks very low (should be higher), but we found out, that when this ratio was higher, lot of domains were banned too quickly. Complex websites contain lot of sections with categories, tags, archives, list of articles by date, author, etc. Typical situation was, that the page with connected text was downloaded first, but lot of links from this page links to pages with lists of articles (tages, sections, etc.) without connected text. So this domain got immediately marked for ignoring.

When whole task was processed, domains newly marked as ignored were used to mark all unprocessed URLs in database as invalid (and therefore they will not be chosen). Before any URL was added to the database, it was checked, whether it is from ignored TLD or domain.

This filtering speeds up processing twice.

[44] `https://easylist.adblockplus.org/en/`

crawler.pl

The script `crawler.pl` is responsible for downloading web pages. we used CPAN package LWPx::ParanoidAgent for downloading web pages. Downloading of URL consist of several steps. The HTTP Header is read and HTTP Status code and mime-type are extracted. Only pages with mime-type text/html and status code 2XX are processed further. In the next step, the content charset is retrieved. A complete webpage is converted into utf-8 encoding with package Text::Iconv. If conversion fails or empty content is returned, then processing of this URL is stopped. The converted webpage is normalized by tidy[45] with options `-utf8 -asxml -b -q`.

parser.pl

The script `parser.pl` is used for extracting texts and links from web pages. We used CPAN module HTML::Parser for parsing. The parser extracts only texts of paragraph (inside elements `<p>`). The text from the paragraph is added to the result, if it is considered as valid. A valid paragraph:

- contains at least 8 words - ommits poorly written lists and headers: `<p>Item 1</p><p>Item 2</p><Item 3</p>`.
- contains less than twice more words than links - ommits menus `<p><a>Menu 1</a><br><a><Menu 2</a></p>`.
- Does not contains too much punctuation (less than 66% of words).

All these constants were empirically selected during initial phases of development.

During testing, we found out that the amount of poorly written web pages is much higher, than we expected. Therefore, usually only a very small amount of text was selected. This was caused by using `div` tags instead of `p` or by dividing long texts just by `br` tags. When the extracted text was smaller than 20% of complete webpage size, then all `div` and `td` tags were treated as `p`.

detector.pl

The script `detector.pl` is responsible for the language detection of downloaded texts. At the beginning, it receives the language model from the master. Only

[45] `http://tidy.sourceforge.net/`

texts with at least 50 words (or 300 characters) are identified. Language identification is described in Section 3.3.

Data Format

For each language several files were created. File `web-texts.tar.gz` contains extracted archive and main result file `res.tar`, which is tar archive of all downloaded tasks (3.5.1).

3.5.2 Compiling Corpus

This second phase took place few months after the first one. Meanwhile the system for language identification and text extraction was improved. In the first phase we were also discarding useful texts (when language X was downloaded, then texts in other languages were discarder). For this reason we decided to parse and detect all downloaded files.

Text Extraction

For corpus compilation we used our cluster and following method.

The main result file for was each language (3.5.1) was extracted on a local disc on a cluster node. This approach allowed us to process more languages in the same time, because we eliminated using shared network storage. When the result file was extracted, all tasks were divided into three groups, which were processed simultaneously (i.e., in parallel).

All tasks in a single group were processed in serial as follows:

- Extract task
- Read log information (protocol) about downloaded pages (URL, HTTP Statsu, size, ...)
- Process all successfully downloaded pages (HTTP Status is equal to 200)
- If the URL was already processed, then skip it.
- Extract text and identify language.
- Store extracted texts gzipped in memory with metadata about language and other information (size, URL, ...)

When all tasks from the single group were extracted, then all texts and metadata for each language were stored in the result directory. So when all languages and all groups were processed, then the result folder contains 100 folders for each language and each such folder contained at most 300 files (3 times 100) files with texts and same amount of files with metadata.

Duplicity Detection

The next step in corpus compilation was duplicity detection. Duplicity detection was performed on two levels - URLs and paragraphs.

We decided to detect duplicity on paragraph level. Duplicity detection on paragraph level is more fine-grained in comparison with document level because it does not throw away whole document if it contains duplicate passage. There are at least three reasons for such approach - spam, common passages, and incorrectly detected boilerplate code.

The spam problem is caused by fact, that a good position in search engine results is crucial for business success. There are thousands of pages trying to sell the same product, but users usually click only on the top few links. Therefore, spammers are trying to manipulate with the search engine indexing (this technique is called spamdexing[46]). They build link farms[47] or scaper sites[48] — automatically generated websites that are tightly-knit pages referring to each other. Content is typically generated from Wikipedia or other publicly available resources. To trick the search engines, these websites do not contain exact copies of original texts, but rather only mixed fractions. These spamdexing techniques may cause problems during crawling. If breadth-first approach is used, then the crawler may get stucked in this farm. It may also fool the duplicity detection. Another technique used by spammers, is spamming blogs,[49] where bots comment blog spots. These comments contain links to the spammers' website to increase its popularity. Projects like Honey Pot[50] or Akismet[51] are catching millions of spam comments every day. Spam in comments may also be the source of duplicities and therefore decrease the corpus quality. When a blogger writes a spot on his/her

[46] http://en.wikipedia.org/wiki/Spamdexing
[47] http://en.wikipedia.org/wiki/Link_farm
[48] http://en.wikipedia.org/wiki/Scraper_site
[49] http://en.wikipedia.org/wiki/Spam_in_blogs
[50] http://www.projecthoneypot.org/statistics.php
[51] http://akismet.com/

blog in language X, the text is valuable for the corpus. Later, when a few spam comments are attached, this article will still be identified as language X, but it will not be so valuable, because it will also contain some English sentences. When many such articles are added, the same comments may be presented many times.

The common passages problem is caused by writers that need to define terms in their articles. The general approach is using definition from the Wikipedia.[52]

And the last reason is removing boilerplate code, which will be repeated on every page from single website.

After the duplicity reduction step contains only unique paragraphs.

3.6 Corpus Distribution

The corpus is distributed in form of gzipped text for each language. These files may be downloaded directly from the website `http://ufal.mff.cuni.cz/~majlis/w2c/`.

Data may be used for academic research and commercial usage is subject of separate negotiations and a written contract.

There are available following data for each language:

- web corpus
- wiki corpus
- corpus statistics for both corpora such as word and sentence length, conditional entropy and perplexity, and most frequent characters and words
- 1000 most frequent 1-5-grams for both corpora

[52] `https://www.google.com/search?q=%22The+Internet+is+a+global+system+of+interconnected+c`

4. Results

This chapter describes the amount and properties of collected data. At the beginning of this chapter, the W2C Wiki Corpus (4.1) size is presented. Then the results for the W2C Web Corpus 4.2 and its comparison with the Wiki Corpus are presented.

Tables are sorted alphabetically according to the ISO 639-3 code. All used codes are in Table A. The highest five values in each column are printed $\overline{\textit{overlined}}$ and the lowest five are printed *underlined*.

4.1 W2C Wiki Corpus

The W2C Wiki Corpus contains 106 languages with total size of 8.53 GB. Detailed information about sizes for particular languages are presented in Table 4.1. These sizes are also depicted in Figure 4.1.

The biggest outlier is the Kannada language (kan) which with just 10 thousand articles has 120 MB. It seems that many articles are complete translations of articles from English Wikipedia[53]. The Kannada language is written in the Kannada script which consumes 3 bytes per character[54], so it may contains up to 3 times less characters. A similar explanation also applies for languages Thai (tha), Gujarati (guj) and Burmese (mya).

4.2 W2C Web Corpus

The W2C Web Corpus was the main goal of this project. Methods used for its construction are described in Section 3.5. During downloading phase more than 4 TB and 100 million web pages were downloaded. When error pages and duplicate content was filtered, then only 32 millions unique URLs with total raw size 2 TB were used.

The W2C Web Corpus contains 106 languages with total size of 54.77 GB. Detailed information about sizes for particular languages are presented in Table 4.2

[53] e.g. `http://kn.wikipedia.org/wiki/%E0%B2%B5%E0%B3%87%E0%B2%B2%E0%B3%8D%E0%B2%B8%E0%B3%8` — and other articles about countries

[54] `http://www.unicode.org/charts/PDF/U0C80.pdf` — Kannada Script

ISO	Bytes	Words	ISO	Bytes	Words	ISO	Bytes	Words
afr	28	4.50	heb	234	4.43	oci	12	1.94
als	16	2.45	hif	0	.16	pam	2	.37
ara	183	1.72	hin	209	1.81	pol	137	17.80
arg	16	2.74	hrv	98	14.36	por	165	25.49
arz	9	.11	hun	160	19.52	que	1	.22
ast	12	1.86	hye	22	.34	ron	123	18.07
aze	61	6.20	ina	3	.59	rus	350	5.67
bel	46	.92	ind	95	13.13	sah	4	.10
ben	51	.27	isl	25	3.30	scn	6	.95
bos	33	4.95	ita	211	31.84	sco	6	1.07
bpy	27	.36	jav	10	1.44	slk	78	10.53
bre	19	3.27	jpn	267	.91	slv	73	10.96
bul	169	3.04	kan	120	1.06	spa	282	45.27
cat	134	21.95	kat	107	1.37	sqi	39	6.14
ces	120	16.14	kaz	103	1.95	srp	144	2.94
cos	1	.18	kor	138	3.06	sun	7	1.14
cym	18	3.11	kur	8	1.28	swa	12	2.00
dan	84	12.81	lat	19	2.63	swe	109	15.72
deu	342	45.65	lav	41	5.19	tam	148	1.31
ell	205	2.88	lim	7	1.13	tat	10	.28
eng	429	69.32	lit	69	8.42	tel	130	1.41
epo	64	9.86	lmo	8	1.41	tgk	4	.10
est	71	8.84	ltz	17	2.52	tgl	14	2.21
eus	81	10.43	mal	86	.67	tha	228	1.31
fao	2	.40	mar	24	.46	tur	107	12.65
fas	137	1.16	mkd	107	1.58	ukr	214	3.94
fin	127	13.86	mlg	11	1.54	urd	25	.12
fra	273	41.17	mon	14	.23	uzb	3	.40
fry	19	3.20	mri	1	.31	vec	4	.83
gla	3	.50	msa	72	9.99	vie	136	21.90
gle	12	1.97	mya	51	.17	war	1	.20
glg	90	14.11	nds	20	3.28	yid	13	.18
glk	1	.03	nep	23	.25	yor	1	.23
guj	64	.72	nld	145	22.42	zho	164	.76
hat	6	1.15	nno	61	9.54			
hbs	82	12.07	nor	98	15.04			

Table 4.1: W2C Wiki Corpus – size

Columns — *ISO*: ISO 639-3 code, *Bytes*: size in MB, *Words*: number of words in millions.

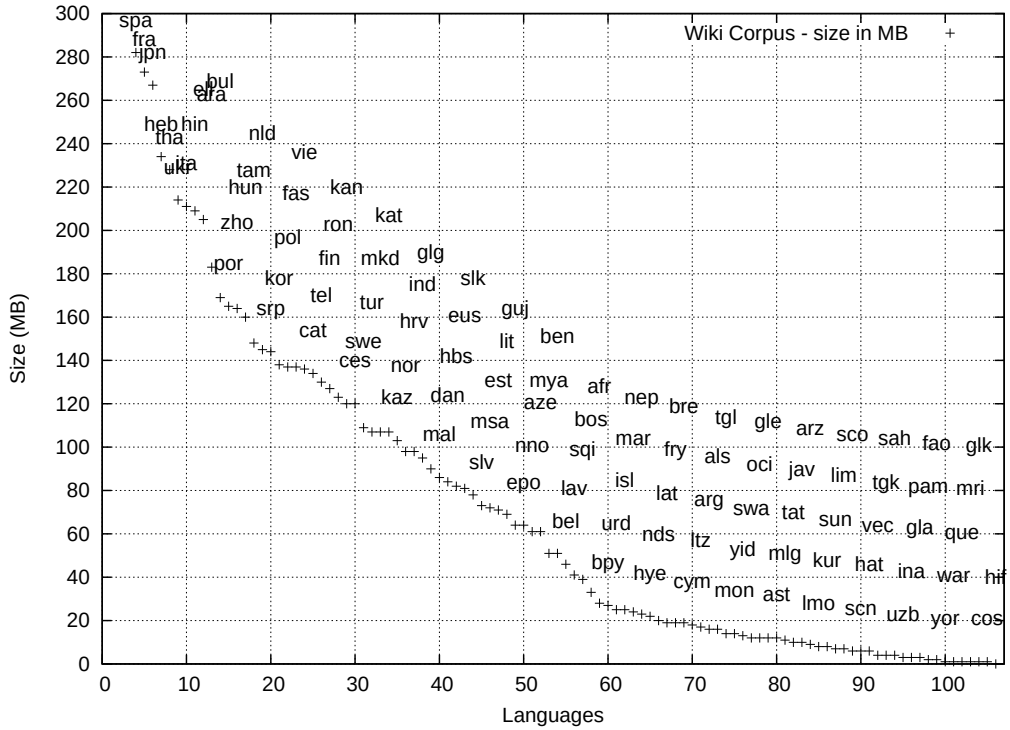

Figure 4.1: W2C Wiki Corpus — size in MB

Languages are sorted according to their size in the W2C Wiki Corpus.

and visualized in Figure 4.2.

The collected size differs for various languages – for 34 languages more than 640 MB of texts are available, for 72 languages more than 160 languages, and for 100 languages more than 10 MB of texts. More details are presented in Table 4.3.

4.3 Comparing Wiki and Web Corpus

Comparing W2C Wiki Corpus and W2C Web Corpus is one of the possibilities how to check whether reliable data was downloaded. Several different properties may point to a language for which suspicious material was collected.

For comparing Wikipedia and the Internet are used following properties:

- Average Word Length (4.3.1)
- Average Sentence Length (4.3.2
- Conditional Entropy and Perplexity (4.3.3)

The values presented should be used with caution, because their main purpose

ISO	Bytes	Words	ISO	Bytes	Words	ISO	Bytes	Words
afr	455	78.71	heb	618	12.89	oci	71	11.74
als	43	6.88	hif	77	14.80	pam	95	13.94
ara	943	10.99	hin	520	4.77	pol	660	89.17
arg	10	1.63	hrv	690	101.53	por	525	82.20
arz	29	.39	hun	736	91.95	que	4	.51
ast	60	9.52	hye	353	6.16	ron	980	155.12
aze	291	32.40	ina	27	4.22	rus	479	10.32
bel	650	13.51	ind	993	143.08	sah	344	5.77
ben	583	4.15	isl	562	80.97	scn	19	2.95
bos	799	124.63	ita	854	131.78	sco	35	5.79
bpy	42	.33	jav	12	1.72	slk	562	78.51
bre	37	6.75	jpn	2283	39.03	slv	574	89.07
bul	670	13.34	kan	398	5.06	spa	1401	228.06
cat	578	95.53	kat	690	9.82	sqi	507	80.78
ces	1035	144.04	kaz	507	8.82	srp	845	16.90
cos	20	2.24	kor	554	11.77	sun	4	.53
cym	251	42.85	kur	306	46.16	swa	232	35.47
dan	491	77.71	lat	233	32.85	swe	610	95.30
deu	699	99.03	lav	1055	129.72	tam	1125	11.45
ell	1167	18.56	lim	20	3.33	tat	130	2.39
eng	4601	759.48	lit	734	92.64	tel	465	5.86
epo	229	36.69	lmo	29	4.91	tgk	342	5.47
est	612	81.62	ltz	81	12.90	tgl	283	47.32
eus	499	64.71	mal	900	8.30	tha	2199	14.90
fao	102	14.46	mar	880	10.51	tur	879	105.74
fas	892	8.73	mkd	639	11.91	ukr	873	15.95
fin	833	94.12	mlg	58	8.73	urd	569	3.10
fra	802	123.91	mon	754	14.15	uzb	185	22.28
fry	72	12.27	mri	78	14.48	vec	13	2.30
gla	38	6.41	msa	503	70.33	vie	530	87.25
gle	541	86.80	mya	1052	6.21	war	4	.68
glg	225	35.80	nds	24	3.85	yid	125	2.06
glk	4	.11	nep	631	4.47	yor	10	.32
guj	521	7.13	nld	808	129.06	zho	20	.27
hat	79	14.97	nno	46	7.31			
hbs	732	113.06	nor	677	108.87			

Table 4.2: W2C Web Corpus – size

Columns — *ISO*: ISO 639-3 code, *Bytes*: size in MB, *Words*: number of words in millions.

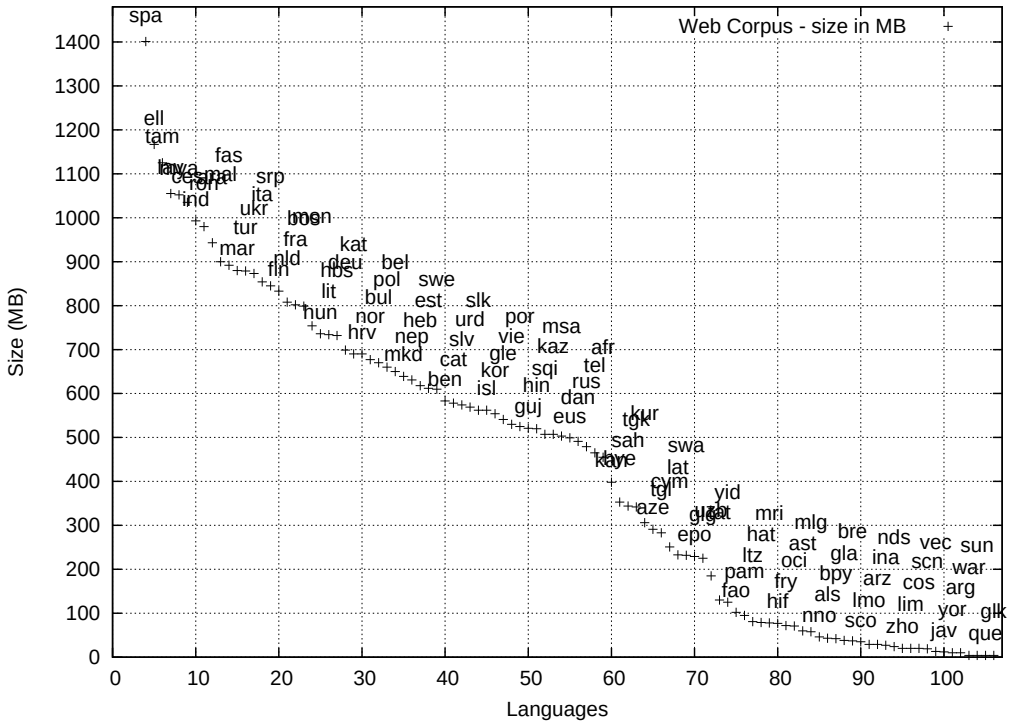

Figure 4.2: W2C Web Corpus — size in MB

Languages are sorted according to their size in the W2C Web Corpus.

Size	Languages
> 10	100
> 20	94
> 40	87
> 80	77
> 160	72
> 320	63
> 640	34

Table 4.3: Number of *Languages* with more texts than *Size* MB.

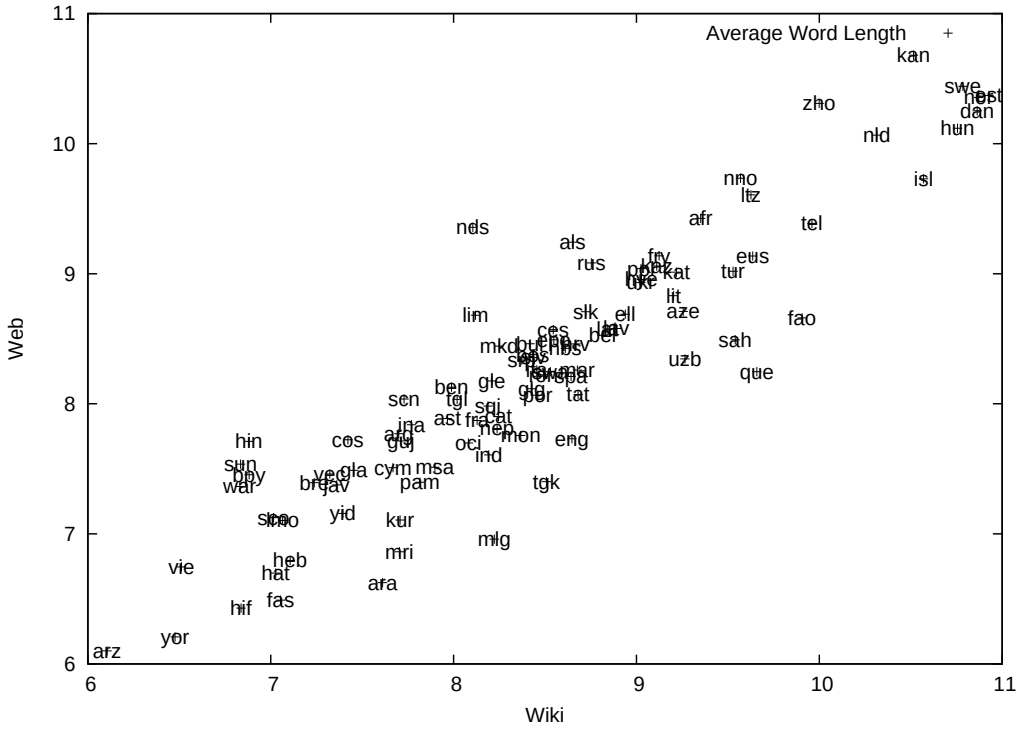

Figure 4.3: Wiki vs Web — average word length

Raw data are in Table B.1

was only the comparison of both corpora. The numbers can be significantly changed by different preprocessing.

4.3.1 Average Word Length

The average word length may reveal problems caused by HTML parsing. From the overall statistics shown in Table 4.4 we can assume, that the downloaded data has reasonable quality because ratio between word lengths on Wikipieda and on Internet are around 1.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.3832	0.9481	0.9734	0.9731	1.0051	1.1545

Table 4.4: Wiki vs Web — average word length – ratio

The raw data are presented in Table B.1 and visualized in Figure 4.3. The biggest outlier is Burmese language (mya), which has almost 3 times shorter words on Wikipedia than on Internet.

4.3.2 Average Sentence Length

The average sentence length is also good measure for the text quality, because it could also reveal some errors in removing boiler plate code. The statistics for ratio between Internet and Wikipedia sentence lengths are presented in Table 4.5. As we can see median and means are also around 1 so it means that many languages are processed correctly.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.4278	0.8632	0.9601	1.5592	1.0681	59.5807

Table 4.5: Wiki vs Web — average word length – ratio

Row data is presented in Table B.2 and visualized in Figure 4.4.

The biggest outliers in this metric is again Burmese language (mya), which has average sentence length on Wikipedia allmost 1586 words whereas on Internet only 27. Checking any page on Burmese Wikipedia[55] reveals that it does not contain any dot, so whole paragraph is treated as a single sentence, whereas extracted segments from the Internet are much shorter and this is causing the difference.

4.3.3 Conditional Entropy and Conditional Perplexity

The conditional entropy is another measure for comparing text quality retrieved from Wikipedia and from Internet. Overall statistics for ratios are presented in Table 4.6. The ratio between Wikipedia and Internet is in average 0.88, which reflects the fact, that data available on Internet has higher variety.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.2641	0.8140	0.8939	0.8871	0.9615	2.6049

Table 4.6: Wiki vs Web — average word length – ratio

Raw data is presented in Table B.3 and visualized in Figure 4.5.

The conditional perplexity is presented in Table B.4 and visualized in Figure 4.6.

[55] `http://my.wikipedia.org/wiki/`

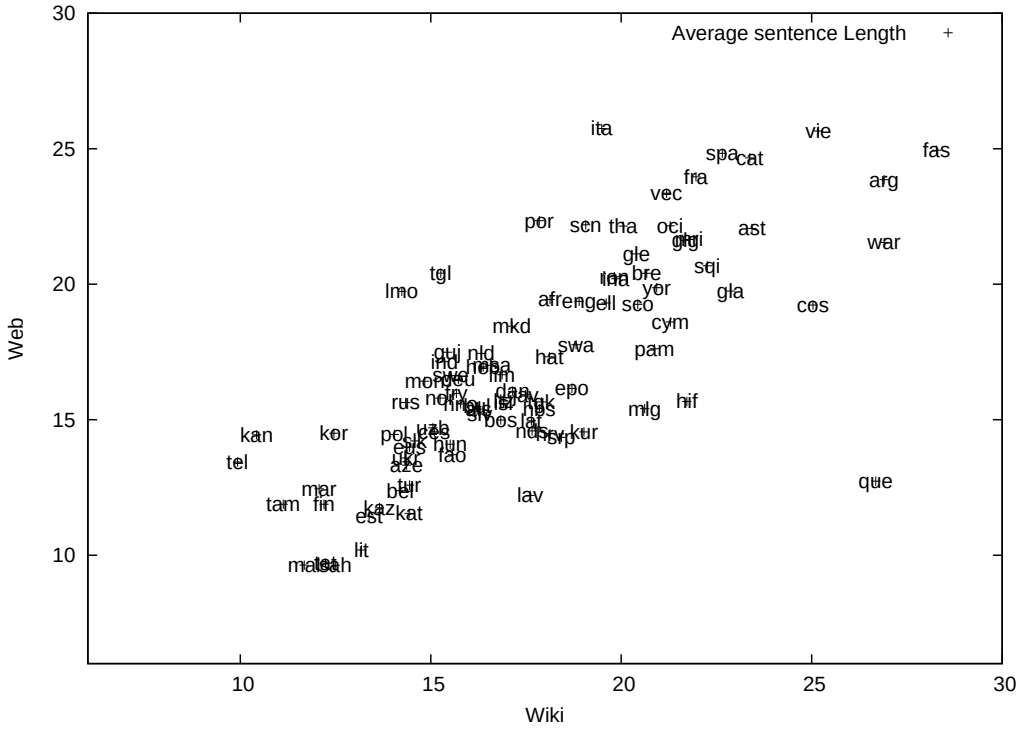

Figure 4.4: Wiki vs Web — average sentence length

Raw data are in Table B.2

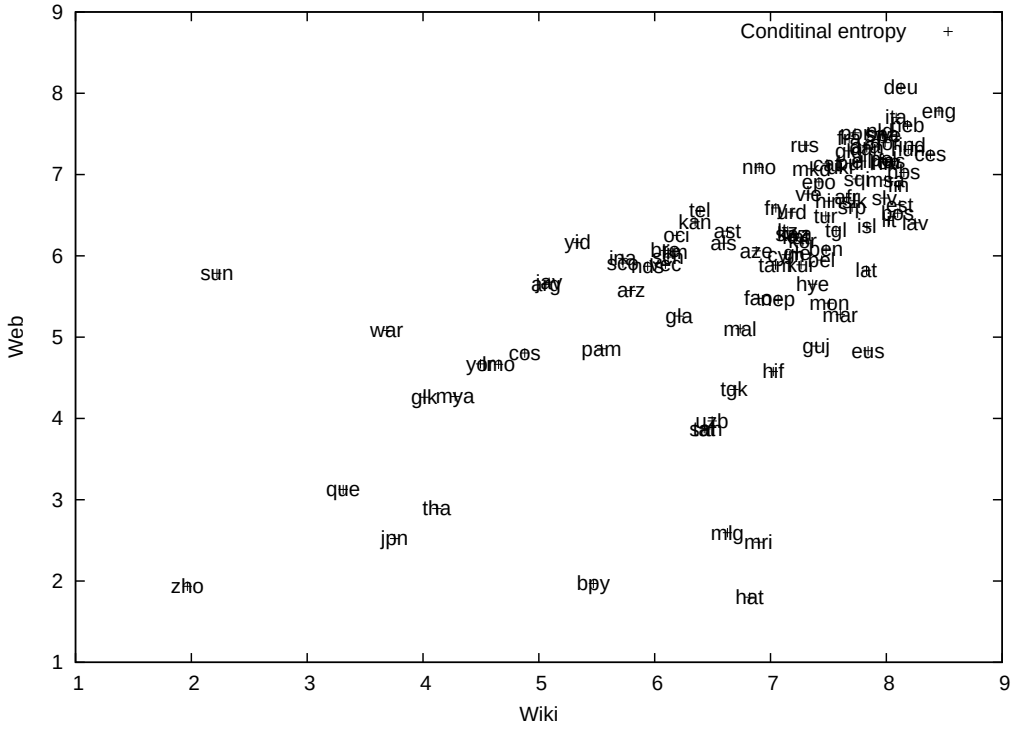

Figure 4.5: Wiki vs Web — conditional entropy

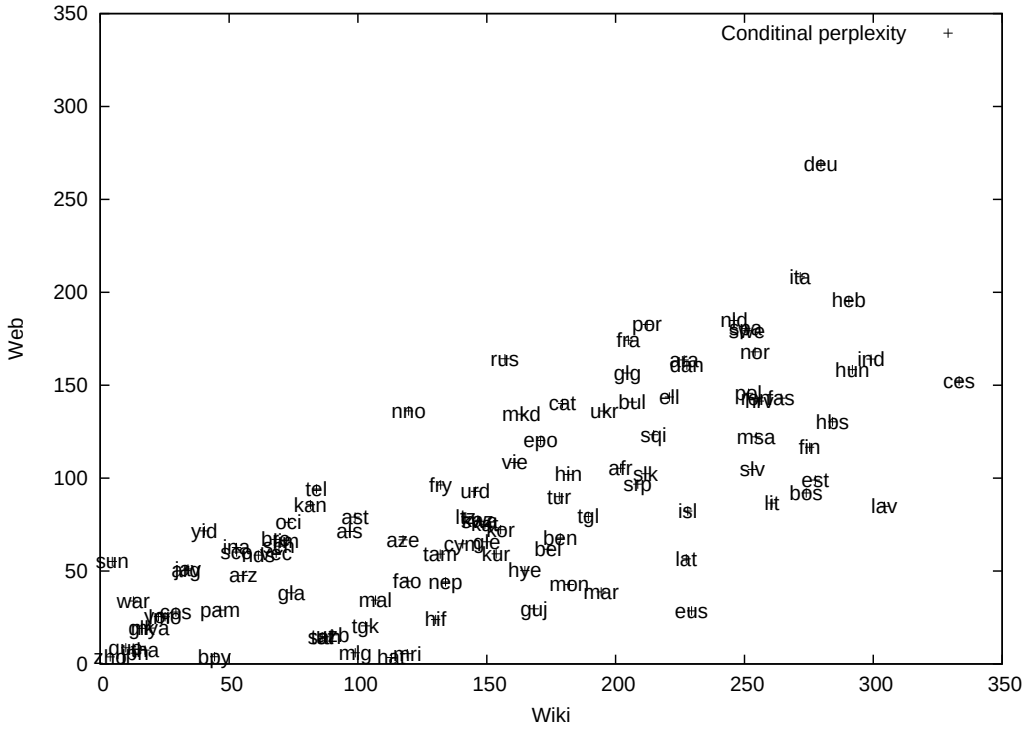

Figure 4.6: Wiki vs Web — conditional perplexity

4.3.4 Conclusions

All outliers have in common, that they are either from minor languages, such as Maori (mri), Malagasy (mlg), for which low quality texts were collected, or they are written in non-latin scripts, such as Japanese (jpn), Chinese (zho), Nepali (nep), Burmese language (mya), which are sensitive to preprocessing.

When different clustering algorithms were applied, then languages in same clusters does not have too much common properties.

5. Conclusions

The W2C Web Corpus consists of 106 languages. For the purpose of corpus construction tools for collecting metadata, building corpus from Wikipedia, language identification, crawling, duplicity reduction, and statistical analysis were developed.

The language metadata is automatically extracted from Ethnologue and Wikipedia and stored in the database. The collected metadata is used by all the components.

Wikipedia was used as the source for the initial corpus. The Wiki Corpus was constructed from Wikipedias with at least 5 thousand articles. The Wiki Corpus contains 20 thousand articles (or as many as available) for 106 languages. This corpus served for training and testing of a language identifier, as well as a baseline for comparison with the web corpus.

The raw corpus of downloaded data contained at least 10 MB for 100 languages included at that time and for 77 of them more than 80 MB. The total corpus size is almost 55 GB of texts.

Both corpora were statistically analysed and compared.

Downloading hundreds of languages would require collecting initial corpus for this amount of languages, which are not easily accessible. If this initial corpora would be available, highly specialized language identifier for each language would be necessary, because only very short text fragments would be analysed. And even if this identifier would be available, it still could not be possible to automatically download the texts, because they are not available on-line.

All downloaded data, more than 4.5 TB, were preserved, so that they can be investigated further and more information about real language usage can be revealed, such as distribution of encodings or scripts for each language. Different tools for text extraction, language identification and duplicity detection may be plugged-in. If the text extractor could extract texts segments instead of complete pages, it would be possible to increase corpus size for minor languages. A different set-up of existing tools allows constructing corpora for many purposes, from the hight quality ones for manual usage to the low quality ones for machine processing. Also, a specialized single topic corpus could be compiled.

Also, many partial topics can be investigated in a more detailed way. For example

the language identification problem, where dozens of parameters and methods combinations were ad-hoc tested, requires more rigorous approach. The text extraction problem could be studied as a complex problem together with duplicity reduction. Where a much simpler extractor does not remove all boilerplate code, but with duplicity reduction on line level, this boilerplate code is removed. All these methods could also be investigated from a performance view, where simpler methods could save weeks of computation for the cost of slightly decreased quality.

The W2C Corpus is a unique data source for linguists, because it outclasses all published works both in the size of collected material and the number of covered languages. The collected data may be used for comparative analysis of related languages, building language models for various applications such as machine translation, speech recognition, spell checking, etc.

A. List of Languages

All information are automatically extracted from ethnologue[56].

Column — *Lang*: ISO 639-3 code, *Name*: language name, *Pop*: population in thousands, *Type*: Living, Extinct, Ancient, Historic, or Constructed, and *Script*: used script

Table A.1: List of Languages

ISO	Name	Pop	Type	Classification
afr	Afrikaans	4934	Liv	Indo-European, Germanic, West
als	Tosk Albanian	3035	Liv	Indo-European, Albanian, Tosk
ara	Arabic	221002	Liv	Afro-Asiatic, Semitic, Central
arg	Aragonese	2000	Liv	Indo-European, Italic, Romance
arz	Egyptian Arabic	53990	Liv	Afro-Asiatic, Semitic, Central
ast	Asturian	125	Liv	Indo-European, Italic, Romance
aze	Azerbaijani	19147	Liv	Altaic, Turkic, Southern
bel	Belarusian	8618	Liv	Indo-European, Slavic, East
ben	Bengali	181272	Liv	Indo-European, Indo-Iranian, Indo-Aryan
bos	Bosnian	2203	Liv	Indo-European, Slavic, South
bpy	Bishnupriya	115	Liv	Indo-European, Indo-Iranian, Indo-Aryan
bre	Breton	500	Liv	Indo-European, Celtic, Insular
bul	Bulgarian	9097	Liv	Indo-European, Slavic, South
cat	Catalan	11530	Liv	Indo-European, Italic, Romance
ces	Czech	9490	Liv	Indo-European, Slavic, West
cos	Corsican	402	Liv	Indo-European, Italic, Romance
cym	Welsh	537	Liv	Indo-European, Celtic, Insular
dan	Danish	5581	Liv	Indo-European, Germanic, North
deu	German	90294	Liv	Indo-European, Germanic, West
ell	Modern Greek	13084	Liv	Indo-European, Greek, Attic
eng	English	328008	Liv	Indo-European, Germanic, West
epo	Esperanto	0	Con	Constructed language
est	Estonian	1048	Liv	Uralic, Finnic
eus	Basque	658	Liv	Basque
fao	Faroese	48	Liv	Indo-European, Germanic, North
fas	Persian	31381	Liv	Indo-European, Indo-Iranian, Iranian
fin	Finnish	5009	Liv	Uralic, Finnic
fra	French	67838	Liv	Indo-European, Italic, Romance
fry	Western Frisian	467	Liv	Indo-European, Germanic, West
gla	Scottish Gaelic	66	Liv	Indo-European, Celtic, Insular

Continued on Next Page...

[56] `http://ethnologue.org`

ISO	Name	Pop	Type	Classification
gle	Irish	391	Liv	Indo-European, Celtic, Insular
glg	Galician	3185	Liv	Indo-European, Italic, Romance
glk	Gilaki	3270	Liv	Indo-European, Indo-Iranian, Iranian
guj	Gujarati	46493	Liv	Indo-European, Indo-Iranian, Indo-Aryan
hat	Haitian	7701	Liv	Creole, French based
hbs	Serbo-Croatian	16351	Liv	Indo-European, Slavic, South
heb	Hebrew	5316	Liv	Afro-Asiatic, Semitic, Central
hif	Fiji Hindi	380	Liv	Indo-European, Indo-Iranian, Indo-Aryan
hin	Hindi	181676	Liv	Indo-European, Indo-Iranian, Indo-Aryan
hrv	Croatian	5546	Liv	Indo-European, Slavic, South
hun	Hungarian	12501	Liv	Uralic
hye	Armenian	6376	Liv	Indo-European, Armenian
ina	Interlingua	0	Con	
ind	Indonesian	23187	Liv	Austronesian, Malayo-Polynesian, Malayo-
isl	Icelandic	238	Liv	Indo-European, Germanic, North
ita	Italian	61696	Liv	Indo-European, Italic, Romance
jav	Javanese	84608	Liv	Austronesian, Malayo-Polynesian, Javanes
jpn	Japanese	122080	Liv	Japonic
kan	Kannada	35327	Liv	Dravidian, Southern, Tamil-Kannada
kat	Georgian	4255	Liv	Kartvelian, Georgian
kaz	Kazakh	8331	Liv	Altaic, Turkic, Western
kor	Korean	66305	Liv	Language isolate
kur	Kurdish	16025	Liv	Indo-European, Indo-Iranian, Iranian
lat	Latin	0	Anc	Indo-European, Italic, Latino-Faliscan
lav	Latvian	1504	Liv	Indo-European, Baltic, Eastern
lim	Limburgan	1300	Liv	Indo-European, Germanic, West
lit	Lithuanian	3154	Liv	Indo-European, Baltic, Eastern
lmo	Lombard	9133	Liv	Indo-European, Italic, Romance
ltz	Luxembourgish	320	Liv	Indo-European, Germanic, West
mal	Malayalam	35893	Liv	Dravidian, Southern, Tamil-Kannada
mar	Marathi	68061	Liv	Indo-European, Indo-Iranian, Indo-Aryan
mkd	Macedonian	2113	Liv	Indo-European, Slavic, South
mlg	Malagasy	14736	Liv	Austronesian, Malayo-Polynesian, Greater
mon	Mongolian	5720	Liv	Altaic, Mongolic, Eastern
mri	Maori	60	Liv	Austronesian, Malayo-Polynesian, Central
msa	Malay	39144	Liv	Austronesian, Malayo-Polynesian, Malayo-
mya	Burmese	32319	Liv	Sino-Tibetan, Tibeto-Burman, Lolo-Burmes
nds	Low German	1	Liv	Indo-European, Germanic, West
nep	Nepali	13875	Liv	Indo-European, Indo-Iranian, Indo-Aryan
nld	Dutch	21730	Liv	Indo-European, Germanic, West
nno	Norwegian Nynorsk	0	Liv	
nor	Norwegian	4640	Liv	Indo-European, Germanic, North

Continued on Next Page...

ISO	Name	Pop	Type	Classification
oci	Occitan	2048	Liv	Indo-European, Italic, Romance
pam	Pampanga	1905	Liv	Austronesian, Malayo-Polynesian, Philipp
pol	Polish	39990	Liv	Indo-European, Slavic, West
por	Portuguese	177981	Liv	Indo-European, Italic, Romance
que	Quechua	10098	Liv	Quechuan, Quechua II, C
ron	Romanian	23351	Liv	Indo-European, Italic, Romance
rus	Russian	143553	Liv	Indo-European, Slavic, East
sah	Yakut	443	Liv	Altaic, Turkic, Northern
scn	Sicilian	4830	Liv	Indo-European, Italic, Romance
sco	Scots	200	Liv	Indo-European, Germanic, West
slk	Slovak	5019	Liv	Indo-European, Slavic, West
slv	Slovenian	1909	Liv	Indo-European, Slavic, South
spa	Spanish	328518	Liv	Indo-European, Italic, Romance
sqi	Albanian	5825	Liv	Indo-European, Albanian, Gheg
srp	Serbian	7020	Liv	Indo-European, Slavic, South
sun	Sundanese	34000	Liv	Austronesian, Malayo-Polynesian, Malayo-
swa	Swahili	730	Liv	Niger-Congo, Atlantic-Congo, Volta-Congo
swe	Swedish	8311	Liv	Indo-European, Germanic, North
tam	Tamil	65675	Liv	Dravidian, Southern, Tamil-Kannada
tat	Tatar	6496	Liv	Altaic, Turkic, Western
tel	Telugu	69758	Liv	Dravidian, South-Central, Telugu
tgk	Tajik	4457	Liv	Indo-European, Indo-Iranian, Iranian
tgl	Tagalog	23853	Liv	Austronesian, Malayo-Polynesian, Philipp
tha	Thai	20362	Liv	Tai-Kadai, Kam-Tai, Be-Tai
tur	Turkish	50750	Liv	Altaic, Turkic, Southern
ukr	Ukrainian	37029	Liv	Indo-European, Slavic, East
urd	Urdu	60586	Liv	Indo-European, Indo-Iranian, Indo-Aryan
uzb	Uzbek	20250	Liv	Altaic, Turkic, Eastern
vec	Venetian	6230	Liv	Indo-European, Italic, Romance
vie	Vietnamese	68634	Liv	Austro-Asiatic, Mon-Khmer, Viet-Muong
war	Waray	2570	Liv	Austronesian, Malayo-Polynesian, Philipp
yid	Yiddish	2255	Liv	Indo-European, Germanic, West
yor	Yoruba	19380	Liv	Niger-Congo, Atlantic-Congo, Volta-Congo
zho	Chinese	1212515	Liv	Sino-Tibetan, Chinese

B. Wiki vs Web

This appendix contains raw data for comparing the Wiki Corpus and the W2C Corpus.

- Average Word Length (B.1)
- Average Sentence Length (B.2
- Conditional Entropy (B.3)
- Conditional Perplexity (B.4)

ISO	Web	Wiki	R	ISO	Web	Wiki	R	ISO	Web	Wiki	R
afr	9.35	9.43	1.01	heb	7.11	6.80	0.96	oci	8.08	7.70	0.95
als	8.65	9.24	1.07	hif	6.84	6.43	0.94	pam	7.82	7.40	0.95
ara	7.61	6.62	0.87	hin	6.88	7.71	1.12	pol	9.03	9.04	1.00
arg	7.70	7.77	1.01	hrv	8.67	8.46	0.98	por	8.46	8.07	0.95
arz	6.11	6.10	1.00	hun	10.76	10.12	0.94	que	9.66	8.25	0.85
ast	7.97	7.89	0.99	hye	9.03	8.96	0.99	ron	8.49	8.22	0.97
aze	9.26	8.71	0.94	ina	7.77	7.84	1.01	rus	8.75	9.08	1.04
bel	8.81	8.53	0.97	ind	8.19	7.61	0.93	sah	9.54	8.49	0.89
ben	7.99	8.13	1.02	isl	10.57	9.73	0.92	scn	7.73	8.04	1.04
bos	8.43	8.38	0.99	ita	8.46	8.26	0.98	sco	7.01	7.12	1.02
bpy	6.88	7.46	1.08	jav	7.36	7.38	1.00	slk	8.72	8.71	1.00
bre	7.24	7.39	1.02	jpn	14.27	14.89	1.04	slv	8.44	8.36	0.99
bul	8.42	8.46	1.01	kan	10.52	10.68	1.02	spa	8.64	8.21	0.95
cat	8.25	7.90	0.96	kat	9.22	9.01	0.98	sqi	8.19	7.98	0.97
ces	8.54	8.57	1.00	kaz	9.11	9.06	0.99	srp	8.37	8.34	1.00
cos	7.42	7.72	1.04	kor	4.97	4.55	0.92	sun	6.83	7.54	1.10
cym	7.67	7.51	0.98	kur	7.71	7.11	0.92	swa	8.53	8.25	0.97
dan	10.86	10.25	0.94	lat	8.84	8.58	0.97	swe	10.79	10.44	0.97
deu	10.88	11.86	1.09	lav	8.89	8.58	0.97	tam	11.73	11.28	0.96
ell	8.94	8.69	0.97	lim	8.12	8.68	1.07	tat	8.68	8.07	0.93
eng	8.65	7.73	0.89	lit	9.20	8.83	0.96	tel	9.96	9.39	0.94
epo	8.55	8.49	0.99	lmo	7.07	7.10	1.01	tgk	8.51	7.40	0.87
est	10.93	10.37	0.95	ltz	9.63	9.61	1.00	tgl	8.02	8.03	1.00
eus	9.64	9.14	0.95	mal	13.29	12.81	0.96	tha	27.96	31.65	1.13
fao	9.91	8.66	0.87	mar	8.68	8.26	0.95	tur	9.53	9.02	0.95
fas	7.05	6.49	0.92	mkd	8.25	8.44	1.02	ukr	9.02	8.93	0.99
fin	12.56	11.86	0.94	mlg	8.22	6.96	0.85	urd	6.74	5.98	0.89
fra	8.13	7.87	0.97	mon	8.37	7.76	0.93	uzb	9.27	8.35	0.90
fry	9.12	9.14	1.00	mri	7.70	6.86	0.89	vec	7.32	7.46	1.02
gla	7.45	7.49	1.00	msa	7.90	7.51	0.95	vie	6.51	6.75	1.04
gle	8.21	8.18	1.00	mya	15.53	5.95	0.38	war	6.83	7.36	1.08
glg	8.43	8.12	0.96	nds	8.11	9.36	1.15	yid	7.39	7.16	0.97
glk	5.92	5.66	0.95	nep	8.24	7.81	0.95	yor	6.48	6.21	0.96
guj	7.71	7.71	1.00	nld	10.31	10.06	0.98	zho	10.00	10.31	1.03
hat	7.03	6.70	0.95	nno	9.57	9.73	1.02				
hbs	8.61	8.43	0.98	nor	10.87	10.36	0.95				

Table B.1: Wiki vs Web — average word length

ISO	Web	Wiki	R	ISO	Web	Wiki	R	ISO	Web	Wiki	R
afr	18.13	19.43	1.07	heb	16.38	16.95	1.04	oci	21.28	22.15	1.04
als	16.26	15.39	0.95	hif	21.73	15.67	0.72	pam	20.88	17.62	0.84
ara	31.22	26.88	0.86	hin	40.45	48.16	1.19	pol	14.04	14.46	1.03
arg	26.90	23.86	0.89	hrv	18.13	14.49	0.80	por	17.85	22.35	1.25
arz	36.51	26.30	0.72	hun	15.51	14.10	0.91	que	26.68	12.73	0.48
ast	23.43	22.06	0.94	hye	56.72	63.98	1.13	ron	19.82	20.28	1.02
aze	14.36	13.32	0.93	ina	19.86	20.20	1.02	rus	14.34	15.64	1.09
bel	14.20	12.38	0.87	ind	15.37	17.14	1.12	sah	12.49	9.63	0.77
ben	76.97	221.66	2.88	isl	16.91	15.60	0.92	scn	19.07	22.19	1.16
bos	16.84	14.99	0.89	ita	19.49	25.75	1.32	sco	20.44	19.26	0.94
bpy	66.89	33.21	0.50	jav	17.49	15.92	0.91	slk	14.58	14.22	0.98
bre	20.67	20.40	0.99	jpn	80.86	160.35	1.98	slv	16.29	15.21	0.93
bul	16.22	15.48	0.95	kan	10.43	14.44	1.38	spa	22.65	24.83	1.10
cat	23.38	24.66	1.05	kat	14.44	11.54	0.80	sqi	22.26	20.68	0.93
ces	15.09	14.55	0.96	kaz	13.66	11.75	0.86	srp	18.43	14.35	0.78
cos	25.04	19.23	0.77	kor	12.46	14.50	1.16	sun	42.22	18.06	0.43
cym	21.29	18.61	0.87	kur	19.02	14.55	0.77	swa	18.82	17.77	0.94
dan	17.15	16.05	0.94	lat	17.64	14.96	0.85	swe	15.53	16.63	1.07
deu	15.72	16.49	1.05	lav	17.61	12.21	0.69	tam	11.13	11.88	1.07
ell	19.61	19.31	0.98	lim	16.87	16.66	0.99	tat	12.24	9.71	0.79
eng	18.89	19.37	1.03	lit	13.18	10.19	0.77	tel	9.93	13.44	1.35
epo	18.71	16.15	0.86	lmo	14.24	19.74	1.39	tgk	17.90	15.66	0.87
est	13.38	11.44	0.85	ltz	16.92	15.69	0.93	tgl	15.27	20.41	1.34
eus	14.45	13.99	0.97	mal	11.68	9.64	0.83	tha	20.05	22.14	1.10
fao	15.57	13.71	0.88	mar	12.07	12.45	1.03	tur	14.44	12.59	0.87
fas	28.28	24.94	0.88	mkd	17.13	18.45	1.08	ukr	14.35	13.59	0.95
fin	12.20	11.88	0.97	mlg	20.62	15.41	0.75	urd	206.41	338.48	1.64
fra	21.96	23.96	1.09	mon	14.85	16.42	1.11	uzb	15.06	14.70	0.98
fry	15.67	15.97	1.02	mri	21.78	21.65	0.99	vec	21.19	23.36	1.10
gla	22.88	19.75	0.86	msa	16.60	17.01	1.02	vie	25.18	25.65	1.02
gle	20.40	21.12	1.04	mya	26.63	1586.40	59.58	war	26.90	21.54	0.80
glg	21.68	21.62	1.00	nds	17.66	14.59	0.83	yid	41.88	24.45	0.58
glk	41.05	20.95	0.51	nep	89.47	72.37	0.81	yor	20.93	19.87	0.95
guj	15.44	17.49	1.13	nld	16.33	17.44	1.07	zho	36.84	122.78	3.33
hat	18.12	17.32	0.96	nno	15.79	15.58	0.99				
hbs	17.85	15.41	0.86	nor	15.24	15.80	1.04				

Table B.2: Wiki vs Web — average sentence length

ISO	Web	Wiki	R	ISO	Web	Wiki	R	ISO	Web	Wiki	R
afr	7.66	6.72	0.88	heb	8.18	7.61	0.93	oci	6.19	6.25	1.01
als	6.60	6.16	0.93	hif	7.02	4.58	0.65	pam	5.54	4.86	0.88
ara	7.82	7.35	0.94	hin	7.51	6.67	0.89	pol	7.97	7.19	0.90
arg	5.06	5.65	1.12	hrv	8.00	7.14	0.89	por	7.73	7.51	0.97
arz	5.80	5.57	0.96	hun	8.19	7.31	0.89	que	3.31	3.13	0.94
ast	6.63	6.30	0.95	hye	7.36	5.65	0.77	ron	7.99	7.16	0.90
aze	6.88	6.06	0.88	ina	5.72	5.97	1.04	rus	7.30	7.36	1.01
bel	7.44	5.95	0.80	ind	8.22	7.36	0.89	sah	6.44	3.87	0.60
ben	7.48	6.08	0.81	isl	7.83	6.36	0.81	scn	6.12	5.99	0.98
bos	8.10	6.52	0.81	ita	8.09	7.70	0.95	sco	5.73	5.91	1.03
bpy	5.47	1.97	0.36	jav	5.09	5.68	1.12	slk	7.73	6.68	0.86
bre	6.09	6.07	1.00	jpn	3.76	2.52	0.67	slv	7.98	6.71	0.84
bul	7.69	7.14	0.93	kan	6.35	6.42	1.01	spa	7.97	7.50	0.94
cat	7.49	7.13	0.95	kat	7.22	6.23	0.86	sqi	7.75	6.95	0.90
ces	8.38	7.25	0.87	kaz	7.20	6.27	0.87	srp	7.70	6.60	0.86
cos	4.88	4.81	0.99	kor	7.28	6.17	0.85	sun	2.22	5.78	2.60
cym	7.14	6.01	0.84	kur	7.26	5.89	0.81	swa	7.20	6.26	0.87
dan	7.83	7.33	0.94	lat	7.83	5.82	0.74	swe	7.97	7.49	0.94
deu	8.13	8.07	0.99	lav	8.25	6.41	0.78	tam	7.04	5.89	0.84
ell	7.79	7.17	0.92	lim	6.17	6.05	0.98	tat	6.43	3.87	0.60
eng	8.45	7.78	0.92	lit	8.03	6.44	0.80	tel	6.39	6.56	1.03
epo	7.42	6.91	0.93	lmo	4.65	4.66	1.00	tgk	6.69	4.36	0.65
est	8.12	6.63	0.82	ltz	7.15	6.31	0.88	tgl	7.57	6.31	0.83
eus	7.84	4.83	0.62	mal	6.74	5.10	0.76	tha	4.12	2.89	0.70
fao	6.90	5.48	0.79	mar	7.60	5.28	0.69	tur	7.48	6.49	0.87
fas	8.05	7.16	0.89	mkd	7.35	7.07	0.96	ukr	7.61	7.09	0.93
fin	8.10	6.87	0.85	mlg	6.63	2.60	0.39	urd	7.19	6.54	0.91
fra	7.68	7.44	0.97	mon	7.51	5.42	0.72	uzb	6.50	3.97	0.61
fry	7.05	6.59	0.94	mri	6.90	2.48	0.36	vec	6.09	5.90	0.97
gla	6.21	5.26	0.85	msa	7.99	6.93	0.87	vie	7.33	6.76	0.92
gle	7.23	6.04	0.84	mya	4.28	4.27	1.00	war	3.69	5.08	1.38
glg	7.68	7.29	0.95	nds	5.94	5.87	0.99	yid	5.34	6.16	1.15
glk	4.01	4.26	1.06	nep	7.07	5.46	0.77	yor	4.50	4.67	1.04
guj	7.40	4.89	0.66	nld	7.94	7.53	0.95	zho	1.97	1.94	0.98
hat	6.82	1.80	0.26	nno	6.90	7.09	1.03				
hbs	8.15	7.03	0.86	nor	7.99	7.39	0.93				

Table B.3: Wiki vs Web — conditional entropy

ISO	Web	Wiki	R	ISO	Web	Wiki	R	ISO	Web	Wiki	R
afr	201.92	105.33	0.52	heb	290.48	195.58	0.67	oci	73.05	76.20	1.04
als	96.71	71.30	0.74	hif	130.16	23.86	0.18	pam	46.57	28.97	0.62
ara	226.63	163.57	0.72	hin	181.66	101.99	0.56	pol	251.55	145.62	0.58
arg	33.38	50.20	1.50	hrv	255.79	141.47	0.55	por	212.27	182.68	0.86
arz	55.63	47.64	0.86	hun	291.87	158.39	0.54	que	9.93	8.74	0.88
ast	98.91	79.02	0.80	hye	164.79	50.23	0.30	ron	254.41	142.79	0.56
aze	117.53	66.75	0.57	ina	52.89	62.63	1.18	rus	157.06	164.22	1.05
bel	173.81	61.71	0.36	ind	299.11	164.06	0.55	sah	87.03	14.63	0.17
ben	178.55	67.76	0.38	isl	227.99	82.25	0.36	scn	69.35	63.40	0.91
bos	273.97	92.04	0.34	ita	271.63	208.39	0.77	sco	52.92	60.17	1.14
bpy	44.36	3.91	0.09	jav	34.05	51.12	1.50	slk	211.64	102.40	0.48
bre	68.16	67.02	0.98	jpn	13.51	5.75	0.43	slv	253.09	104.68	0.41
bul	206.50	140.89	0.68	kan	81.55	85.65	1.05	spa	250.53	180.44	0.72
cat	179.40	140.21	0.78	kat	149.25	75.16	0.50	sqi	214.78	123.31	0.57
ces	333.29	152.32	0.46	kaz	146.56	77.38	0.53	srp	208.62	96.76	0.46
cos	29.40	27.96	0.95	kor	155.38	72.24	0.46	sun	4.66	55.13	11.83
cym	140.65	64.62	0.46	kur	153.66	59.26	0.39	swa	147.20	76.72	0.52
dan	227.65	160.79	0.71	lat	227.41	56.58	0.25	swe	250.99	179.32	0.71
deu	279.65	269.22	0.96	lav	304.30	84.81	0.28	tam	131.94	59.12	0.45
ell	220.84	143.64	0.65	lim	72.21	66.16	0.92	tat	86.11	14.66	0.17
eng	350.81	219.84	0.63	lit	260.65	86.57	0.33	tel	83.93	94.17	1.12
epo	170.92	120.29	0.70	lmo	25.15	25.37	1.01	tgk	102.93	20.49	0.20
est	277.53	98.99	0.36	ltz	141.86	79.19	0.56	tgl	189.45	79.36	0.42
eus	229.47	28.53	0.12	mal	106.83	34.35	0.32	tha	17.37	7.42	0.43
fao	119.14	44.57	0.37	mar	194.42	38.78	0.20	tur	178.16	90.02	0.51
fas	264.21	143.03	0.54	mkd	163.56	134.43	0.82	ukr	195.68	136.10	0.70
fin	275.30	116.60	0.42	mlg	98.94	6.04	0.06	urd	145.61	92.89	0.64
fra	204.93	174.18	0.85	mon	182.07	42.76	0.23	uzb	90.28	15.67	0.17
fry	132.10	96.31	0.73	mri	119.18	5.57	0.05	vec	68.25	59.59	0.87
gla	74.22	38.30	0.52	msa	254.57	122.26	0.48	vie	160.92	108.42	0.67
gle	150.08	65.74	0.44	mya	19.39	19.27	0.99	war	12.89	33.85	2.63
glg	204.60	156.56	0.77	nds	61.37	58.58	0.95	yid	40.41	71.60	1.77
glk	16.11	19.17	1.19	nep	134.00	44.14	0.33	yor	22.56	25.48	1.13
guj	168.38	29.63	0.18	nld	246.03	184.80	0.75	zho	3.91	3.83	0.98
hat	113.01	3.49	0.03	nno	119.66	136.15	1.14				
hbs	284.16	130.30	0.46	nor	254.13	167.94	0.66				

Table B.4: Wiki vs Web — conditional perplexity

Bibliography

[BB01] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 26–33, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[BBFZ09] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43:209–226, 2009. 10.1007/s10579-009-9081-4.

[BGMZ97] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Comput. Netw. ISDN Syst.*, 29:1157–1166, September 1997.

[BK06] Marco Baroni and Adam Kilgarriff. Large linguistically-processed web corpora for multiple languages. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, EACL '06, pages 87–90, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[BRSB00] A. Bharati, K. P. Rao, R. Sangal, and S. M. Bendre. Basic statistical analysis of corpus and cross comparison among corpora. *Technical Report of Indian Institute of Information Technology*, 2000.

[HNP09] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24:8–12, 2009.

[KRPP10] Adam Kilgarriff, Siva Reddy, Jan Pomikálek, and Avinesh PVS. A corpus factory for many languages. In *Language Resources and Evaluation*, 2010.

[Maj11] Martin Majliš. Large multilingual corpus, September 2011.

[RG00] Paul Rayson and Roger Garside. Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing corpora - Volume 9*, WCC '00, pages 1–6, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[Sca07] Kevin P. Scannell. *The Crúbadán Project: Corpus building for under-resourced languages*, volume 4 of *Cahiers du Cental*, pages 5–15. Louvain-la-Neuve, Belgium, 2007.

[Sha06] Serge Sharoff. Creating general-purpose corpora using automated search engine queries. In *WaCky! Working papers on the Web as Corpus. Gedit*, 2006.

[Wyn05] Martin Wynne. *Developing Linguistic Corpora: a Guide to Good Practice*, chapter Archiving, Distribution and Preservation, pages 71–78. Oxford: Oxbow Books, 2005. Available online, Accessed 2011-01-01.

List of Tables

List of Figures

The ÚFAL/CKL Technical Report Series

ÚFAL

ÚFAL (Ústav formální a aplikované lingvistiky; http://ufal.mff.cuni.cz) is the Institute of Formal and Applied linguistics, at the Faculty of Mathematics and Physics of Charles University, Prague, Czech Republic. The Institute was established in 1990 after the political changes as a continuation of the research work and teaching carried out by the former Laboratory of Algebraic Linguistics since the early 60s at the Faculty of Philosophy and later the Faculty of Mathematics and Physics. Together with the "sister" Institute of Theoretical and Computational Linguistics (Faculty of Arts) we aim at the development of teaching programs and research in the domain of theoretical and computational linguistics at the respective Faculties, collaborating closely with other departments such as the Institute of the Czech National Corpus at the Faculty of Philosophy and the Department of Computer Science at the Faculty of Mathematics and Physics.

CKL

As of 1 June 2000 the Center for Computational Linguistics (Centrum komputační lingvistiky; http://ckl.mff.cuni.cz) was established as one of the centers of excellence within the governmental program for support of research in the Czech Republic. The center is attached to the Faculty of Mathematics and Physics of Charles University in Prague.

Technical Reports

The ÚFAL/CKL technical report series has been established with the aim of disseminate topical results of research currently pursued by members, cooperators, or visitors of the Institute. The technical reports published in this Series are results of the research carried out in the research projects supported by the Grant Agency of the Czech Republic, GAČR 405/96/K214 ("Komplexní program"), GAČR 405/96/0198 (Treebank project), grant of the Ministry of Education of the Czech Republic VS 96151, and project of the Ministry of Education of the Czech Republic LN00A063 (Center for Computational Linguistics). Since November 1996, the following reports have been published.

ÚFAL TR-1996-01 Eva Hajičová, *The Past and Present of Computational Linguistics at Charles University*
Jan Hajič and Barbora Hladká, *Probabilistic and Rule-Based Tagging of an Inflective Language – A Comparison*

ÚFAL TR-1997-02 Vladislav Kuboň, Tomáš Holan and Martin Plátek, *A Grammar-Checker for Czech*

ÚFAL TR-1997-03 Alla Bémová at al., Anotace na analytické rovině, *Návod pro anotátory (in Czech)*

ÚFAL TR-1997-04 Jan Hajič and Barbora Hladká, *Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structural Tagset*

ÚFAL TR-1998-05 Geert-Jan M. Kruijff, *Basic Dependency-Based Logical Grammar*

ÚFAL TR-1999-06 Vladislav Kuboň, *A Robust Parser for Czech*

ÚFAL TR-1999-07 Eva Hajičová, Jarmila Panevová and Petr Sgall, *Manuál pro tektogramatické značkování (in Czech)*

ÚFAL TR-2000-08 Tomáš Holan, Vladislav Kuboň, Karel Oliva, Martin Plátek, *On Complexity of Word Order*

ÚFAL/CKL TR-2000-09 Eva Hajičová, Jarmila Panevová and Petr Sgall, *A Manual for Tectogrammatical Tagging of the Prague Dependency Treebank*

ÚFAL/CKL TR-2001-10 Zdeněk Žabokrtský, *Automatic Functor Assignment in the Prague Dependency Treebank*

ÚFAL/CKL TR-2001-11 Markéta Straňáková, *Homonymie předložkových skupin v češtině a možnost jejich automatického zpracování*

ÚFAL/CKL TR-2001-12 Eva Hajičová, Jarmila Panevová and Petr Sgall, *Manuál pro tektogramatické značkování (III. verze)*

ÚFAL/CKL TR-2002-13 Pavel Pecina and Martin Holub, *Sémanticky signifikantní kolokace*

ÚFAL/CKL TR-2002-14 Jiří Hana, Hana Hanová, *Manual for Morphological Annotation*

ÚFAL/CKL TR-2002-15 Markéta Lopatková, Zdeněk Žabokrtský, Karolína Skwarská and Vendula Benešová, *Tektogramaticky anotovaný valenční slovník českých sloves*

ÚFAL/CKL TR-2002-16 Radu Gramatovici and Martin Plátek, *D-trivial Dependency Grammars with Global Word-Order Restrictions*

ÚFAL/CKL TR-2003-17 Pavel Květoň, *Language for Grammatical Rules*

ÚFAL/CKL TR-2003-18 Markéta Lopatková, Zdeněk Žabokrtský, Karolina Skwarska, Václava Benešová, *Valency Lexicon of Czech Verbs VALLEX 1.0*

ÚFAL/CKL TR-2003-19 Lucie Kučová, Veronika Kolářová, Zdeněk Žabokrtský, Petr Pajas, Oliver Čulo, *Anotování koreference v Pražském závislostním korpusu*

ÚFAL/CKL TR-2003-20 Kateřina Veselá, Jiří Havelka, *Anotování aktuálního členění věty v Pražském závislostním korpusu*

ÚFAL/CKL TR-2004-21 Silvie Cinková, *Manuál pro tektogramatickou anotaci angličtiny*

ÚFAL/CKL TR-2004-22 Daniel Zeman, *Neprojektivity v Pražském závislostním korpusu (PDT)*

ÚFAL/CKL TR-2004-23 Jan Hajič a kol., *Anotace na analytické rovině, návod pro anotátory*

ÚFAL/CKL TR-2004-24 Jan Hajič, Zdeňka Urešová, Alevtina Bémová, Marie Kaplanová, *Anotace na tektogramatické rovině (úroveň 3)*

ÚFAL/CKL TR-2004-25 Jan Hajič, Zdeňka Urešová, Alevtina Bémová, Marie Kaplanová, *The Prague Dependency Treebank, Annotation on tectogrammatical level*

ÚFAL/CKL TR-2004-26 Martin Holub, Jiří Diviš, Jan Pávek, Pavel Pecina, Jiří Semecký, *Topics of Texts. Annotation, Automatic Searching and Indexing*

ÚFAL/CKL TR-2005-27 Jiří Hana, Daniel Zeman, *Manual for Morphological Annotation (Revision for PDT 2.0)*

ÚFAL/CKL TR-2005-28 Marie Mikulová a kol., *Pražský závislostní korpus (The Prague Dependency Treebank) Anotace na tektogramatické rovině (úroveň 3)*

ÚFAL/CKL TR-2005-29 Petr Pajas, Jan Štěpánek, *A Generic XML-Based Format for Structured Linguistic Annotation and Its application to the Prague Dependency Treebank 2.0*

ÚFAL/CKL TR-2006-30 Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolařová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, *Annotation on the tectogrammatical level in the Prague Dependency Treebank (Annotation manual)*

ÚFAL/CKL TR-2006-31 Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolařová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Petr Sgall, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, *Anotace na tektogramatické rovině Pražského závislostního korpusu (Referenční příručka)*

ÚFAL/CKL TR-2006-32 Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolařová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Petr Sgall,Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, *Annotation on the tectogrammatical level in the Prague Dependency Treebank (Reference book)*

ÚFAL/CKL TR-2006-33 Jan Hajič, Marie Mikulová, Martina Otradovcová, Petr Pajas, Petr Podveský, Zdeňka Urešová, *Pražský závislostní korpus mluvené češtiny. Rekonstrukce standardizovaného textu z mluvené řeči*

ÚFAL/CKL TR-2006-34 Markéta Lopatková, Zdeněk Žabokrtský, Václava Benešová (in cooperation with Karolína Skwarska, Klára Hrstková, Michaela Nová, Eduard Bejček, Miroslav Tichý) *Valency Lexicon of Czech Verbs. VALLEX 2.0*

ÚFAL/CKL TR-2006-35 Silvie Cinková, Jan Hajič, Marie Mikulová, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jarmila Panevová, Jiří Semecký, Jana Šindlerová, Josef Toman, Zdeňka Urešová, Zdeněk Žabokrtský, *Annotation of English on the tectogrammatical level*

ÚFAL/CKL TR-2007-36 Magda Ševčíková, Zdeněk Žabokrtský, Oldřich Krůza, *Zpracování pojmenovaných entit v českých textech*

ÚFAL/CKL TR-2008-37 Silvie Cinková, Marie Mikulová, *Spontaneous speech reconstruction for the syntactic and semantic analysis of the NAP corpus*

ÚFAL/CKL TR-2008-38 Marie Mikulová, *Rekonstrukce standardizovaného textu z mluvené řeči v Pražském závislostním korpusu mluvené češtiny. Manuál pro anotátory*

ÚFAL/CKL TR-2008-39 Zdeněk Žabokrtský, Ondřej Bojar, *TectoMT, Developer's Guide*

ÚFAL/CKL TR-2008-40 Lucie Mladová, *Diskurzní vztahy v češtině a jejich zachycení v Pražském závislostním korpusu 2.0*

ÚFAL/CKL TR-2009-41 Marie Mikulová, *Pokyny k překladu určené překladatelům, revizorům a korektorům textů z Wall Street Journal pro projekt PCEDT*

ÚFAL/CKL TR-2011-42 Loganathan Ramasamy, Zdeněk Žabokrtský, *Tamil Dependency Treebank (TamilTB) – 0.1 Annotation Manual*

ÚFAL/CKL TR-2011-43 Nguy Giang Linh, Michal Novák, Anna Nedoluzhko, *Coreference Resolution in the Prague Dependency Treebank*

ÚFAL/CKL TR-2011-44 Anna Nedoluzhko, Jiří Mírovský, *Annotating Extended Textual Coreference and Bridging Relations in the Prague Dependency Treebank*

ÚFAL/CKL TR-2011-45 David Mareček, Zdeněk Žabokrtský, *Unsupervised Dependency Parsing*

ÚFAL/CKL TR-2011-46 Martin Majliš, Zdeněk Žabokrtský, *W2C – Large Multilingual Corpus*