Proceedings of the 12th Web as Corpus Workshop, pages 23–32
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Building Web Corpora for Minority Languages

Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén
University of Helsinki, Department of Digital Humanities
firstname.lastname@helsinki.fi

Abstract

Web corpora creation for minority languages that do not have their own top-level Internet domain is no trivial matter. Web pages in such minority languages often contain text and links to pages in the dominant language of the country. When building corpora in specific languages, one has to decide how and at which stage to make sure the texts gathered are in the desired language. In the "Finno-Ugric Languages and the Internet" (Suki) project, we created web corpora for Uralic minority languages using web crawling combined with a language identification system in order to identify the language while crawling. In addition, we used *language set identification* and crowdsourcing before making sentence corpora out of the downloaded texts. In this article, we describe a strategy for collecting textual material from the Internet for minority languages. The strategy is based on the experiences we gained during the Suki project.

Keywords: web corpora, minority languages, language identification, web-crawling

1. Introduction

Web corpus, as we use the term here, refers to a collection of texts that have been acquired from the World Wide Web and been processed into a static corpus (Fletcher (2012), see also Biemann et al. (2013), Schäfer and Bildhauer (2013)). Making a web text corpus in English is fairly straightforward. If one seeds a web crawler with links to pages written in English, one probably ends up having many texts very quickly. With only a moderate amount of post-processing with existing tools, one can have a corpus in English (Tamura et al., 2007). Finding pages in less dominant languages is more difficult as one has to decide how to effectively find the texts in the desired languages.

Building web corpora in minority languages forms a special case within the corpus creation challenge (Barbaresi, 2015, 127–129). When talking about minority languages in this article, we refer to languages that do not have their own national top-level domain (see e.g. Murphy and Stemle (2011), Schulz et al. (2013)). Searching for texts in national top-level domains is a common way of building corpora in specific languages. However, websites containing minority languages are within the same national domain as those in a majority language, hence some kind of language identification is needed. Pages in minority languages often contain links to pages written in the majority language of the country (Arkhangelskiy, 2019). So, even if one seeds a web crawler with links to pages in the desired language, one quickly ends up with many texts in a majority language.

In this paper, we propose a strategy for building web corpora for minority languages. The strategy is based on our experience with building web corpora for Uralic minority languages in the "Finno-Ugric Languages and the Internet" project[1] (Suki) which was active from 2013 to 2019. The minority languages of the Finno-Ugric language group are used mostly in Northern Europe, Estonia and Russia. The written languages use Latin or Cyrillic alphabets with many special characters. A few of the languages have over half a million speakers, but, for example, Votic only has 11 according to Kuznetsova et al. (2015). The aim of the Suki project was to build web corpora for as many of these minority languages as possible in order to facilitate their survival and revival (Jauhiainen et al., 2015a; Jauhiainen et al., 2019a). In our scope, we included the Samojedic languages (within Russia) which, together with the Finno-Ugric languages, form the larger Uralic language group. We used the *Ethnologue* (Simons and Fennig, 2018) together with the ISO-639-3 standard (SIL, 2013) as our source for the division of Uralic languages. Currently, the *Ethnologue* recognizes 38 different Uralic languages.

We start by reviewing previous work on building language-specific web corpora in Section 2. We then describe the various components we used when gathering sentence corpora, that is, corpora composed of sentences instead of entire texts, for 29 small Uralic languages in Section 3. The lessons we gained from doing this are presented in Section 4. Finally, we introduce the outline of our proposed strategy for building web corpora for minority languages in Section 5. We also provide links to the source code of the technical components that we used in our workflow. All our components are published as open source.

2. Previous Research

The ways in which scholars have tried to find pages for building web corpora in various languages vary. There is also variation in how and at which stage the researchers make sure the pages found are in the desired language. Automatic language identification can be performed using methods ranging from a simple function word checkup to deep neural networks. A recent survey by Jauhiainen et al. (2019d) gives a thorough overview of the subject. In this review of previous research, we concentrate on how the web corpora in specific languages have been obtained.

2.1. Pre-Downloaded Web Collections

Instead of collecting the texts from the Internet directly, it is possible to use pre-downloaded collections. Pomikálek et al. (2012) extracted all pages tagged as English from the

[1] `http://suki.ling.helsinki.fi/eng/project.html`

the Lemur project's[2] ClueWeb09 corpus and then used automatic language identification to make sure the texts used were, in fact, written in English. Common Crawl Foundation[3] regularly crawls the Internet and offers the texts it finds for free download. Smith et al. (2013) downloaded document pairs from the Common Crawl corpus for parallel text corpora of several languages and determined the language of a document using language codes present in the URL. Kanerva et al. (2014) used the morphological analyser OMorFi[4] to find Finnish sentences in the Common Crawl corpus. Using the Common Crawl corpus, Schäfer (2016) built corpora in several languages using the texrex tool[5] while Habernal et al. (2016) built corpora in over 50 languages using a java library to determine the language of a text. Panchenko et al. (2018) built a corpus in English using the C4Corpus tool[6] to find relevant pages in the Common Crawl corpus.

2.2. Search Engines

Several scholars have used automatic creation of search queries to find texts in specific languages on the Internet. Ghani et al. (2001) compiled a script called *CorpusBuilder*,[7] which selects terms from two documents, one relevant and the other not, and constructs a query that uses the conjunction of the terms from the relevant document and the negation of the disjunction of the ones from the non-relevant. The top search engine hit for the query is then downloaded, and the document assigned to either the relevant or non-relevant document set. Search engine queries were also used by Sharoff (2006a) in his corpus building strategy and by Ueyama and Baroni (2005) for building a corpus in Japanese.

Baroni and Bernardini (2004) created a toolkit called *BootCaT*,[8] which takes a small set of seed terms and uses queries produced from them to download pages with a search engine. The toolkit was tested by building corpora for English and Italian. BootCaT was also used by Baroni and Ueyama (2004), Sharoff (2006b), and Lyding et al. (2014).

Scannell (2007) built a tool that resembled BootCaT. Depending on the language, the query lists were built with either word lists from a spell checker or word frequency lists. Sometimes language models of trigrams were used to make sure the language was the relevant one. With the tool, Scannell built text corpora for over 400 languages, many of which were under-resourced languages. Arkhangelskiy (2019) used a similar strategy to find texts in seven minority Uralic languages from social media sites. Wagner Filho et al. (2018) also used a toolkit resembling BootCaT called *Web as Corpus Toolkit*[9] for building a web corpus in Brazilian Portuguese.

Schäfer and Bildhauer (2012) recommend that projects wishing to build large web corpora do not use search engine results except as seed URLs for a crawler. The results of their analysis demonstrate that simply downloading the query results with a tool such as BootCaT is not effective enough and that many sites that can be found while crawling the web intensively cannot be found through search engine queries. In addition to this, Sharoff (2006b) and Barbaresi (2013) raise the question of search engines ordering the results according to their "relevance" and the bias this might cause.

2.3. Crawling to Gather Texts

Web crawling is the task of finding large amounts of pages on the web by extracting hyperlinks from already downloaded documents and following them (Olston and Najork, 2010). Web crawlers are used, for example, by search engines to index the web, but also for archiving pages and for data mining. According to Fletcher (2012), it is important to crawl the web if one wants to build web corpora in several languages besides English. Those who have preferred crawling for this have used several different ways of determining what language a page has been written in.

2.3.1. Using URL to Determine a Language

Baykan et al. (2008) wanted to know the language of each page before downloading. They extracted words from URLs and used various machine-learning algorithms to distinguish pages in different languages from each other. Their experiments in various languages showed, however, that English words are prominently present in the URLs of pages in many languages. According to Barbaresi (2013), in case of "lesser-known" languages, language identification of the actual text is necessary even when the words in the URL are used. When searching for pages in Hindi, Priyatam et al. (2012) did prefer to apply a language classifier in addition to the information acquired from the URLs.

2.3.2. Web Page Metadata for Determining Language

The metadata in the HTML source has also been used for determining the language of a page. Somboonviwat et al. (2005) used the information of the pages' charset to determine if they were written in Japanese. Tamura et al. (2007) applied the same method but used TextCat[10] to verify the language of the pages where the charset was found to be UTF-8. They admitted, though, that the metadata check was performed to improve runtime efficiency and that using language identification on pages with other relevant charsets as well would have improved precision.

Identifying the language of a web page by checking the charset in the metadata makes sense only if the language one is interested in uses a special charset. Minority languages are often written using the same encoding as the dominant language of the country they are used in. Furthermore, although many HTML documents contain a language declaration as metadata for the page itself, they are often not used, or used incorrectly (Rehůřek and Kolkus, 2009).

[2] http://www.lemurproject.org/components.php

[3] https://commoncrawl.org

[4] http://flammie.github.io/omorfi/

[5] https://github.com/rsling/texrex

[6] https://dkpro.github.io/dkpro-c4corpus/

[7] http://www.cs.cmu.edu/~TextLearning/corpusbuilder/

[8] https://bootcat.dipintra.it

[9] http://wac-tk.drni.de

[10] https://www.mediawiki.org/wiki/TextCat

2.3.3. Language checking after Crawling

Some scholars have preferred to use some kind of language checking after crawling the web for a corpus. Spoustová and Spousta (2012) and Versley and Panchenko (2012) crawled only some specific, well-chosen sites. Spoustová and Spousta (2012) then used various tools to filter out unwanted texts, whereas Versley and Panchenko (2012) checked the language of each page by inspecting the character encoding and then by using a character trigram-based filter and function words. Emerson and O'Neil (2006) restricted the crawler to accept only pages with metadata language codes indicating the Chinese language. After the crawl, they used the Rosette Language Identifier[11] to detect the language of each page.

Many researchers prefer to crawl national top-level domains where texts in the desired language are believed to be found. Then after the crawl, the language of the pages is verified with various methods. Kornai et al. (2006) applied spell checking to filter out pages that were not in Hungarian. The presence of function words has also been used as a simple form of language identification (Baroni and Kilgariff, 2006; Ferraresi et al., 2008; Baroni et al., 2009), whereas more sophisticated language identifiers were used by Pomikálek et al. (2009) and Schäfer and Bildhauer (2012).

As the .es national domain was very small at the time, Boleda et al. (2006) additionally crawled pages from other domains which were located in Spain in order to build a corpus of Catalan texts. The language of the pages was then identified using a Naive Bayes classifier.

2.3.4. Language Identification during Crawling

Finding web pages dealing with a specific topic is difficult if one just crawls with a standard web crawler (Menczer, 1997; Chakrabarti et al., 1999; Diligenti et al., 2000). A strategy for effectively finding pages on specific topics was proposed by De Bra et al. (1994) and Menczer (1997) but *focused crawling*, an often-used term, was coined by Chakrabarti et al. (1999). A focused crawler assigns a score to the links harvested from a page and the links are handled according to the score they have been assigned thereafter. The idea is that pages on the Internet tend to link to other pages on the same topic (Aggarwal et al., 2001). Somboonviwat et al. (2005) suggested using focused crawling to find pages in specific languages and tested two strategies for doing this. They proposed prioritising links found on pages that had HTML metadata indicating the wanted language and using a threshold for how many irrelevant pages the crawler is allowed to proceed from a relevant one. Schäfer et al. (2014) recommend using the detection of frequent short words and boilerplate to optimise focused web crawling.

In order to prioritise links from a page in a specific language, one needs to check the language while crawling. Many scholars have built web corpora using some kind of focused crawling technique with various language identification methods. Medelyan et al. (2006) used a web crawler named Nutch[12] and identified the language of the pages with TextCat. Mon and Mikami (2011) built their own focused crawler with n-gram based language identification. The links to the subdomain of a page were only added to the outlink queue if the page itself was relevant. Suchomel and Pomikálek (2012) built SpiderLing, a web crawler with inbuilt language models. SpiderLing calculates a yield rate for each page and site. When the yield rate of a site gets too low, it is blacklisted. Barbaresi (2013) used his own crawler to find texts in several different languages and langid.py[13] to identify the language of the crawled pages. He added new links to the download queue only from the relevant pages.

3. Components for building Sentence Corpora for small Uralic Languages

In the Suki project, we started from the premise that crawling the Internet equipped with language identification would give us texts to be processed into corpora in Uralic minority languages. It the end, the main components of our strategy for building web sentence corpora were:

- Acquire the texts using web crawling
- Automatically determine the language using language identification and language set identification
- Verify the automatically identified languages using crowdsourcing
- Tokenise the texts into sentences

3.1. Acquiring texts using web crawling

3.1.1. Choosing a Crawler

Some scholars using web crawlers for collecting pages in specific languages or topics have been concerned that they download too many pages that do not contain what they are looking for (Somboonviwat et al., 2005; Suchomel and Pomikálek, 2012; Schäfer et al., 2014). Schäfer et al. (2014) tried to overcome the problem by collecting seed URLs that were as good quality as possible. Their experiments show that having good-quality seeds is not sufficient when searching for texts in a specific language in national domains containing multiple languages.

Since we were looking for minority languages, we hoped that the relevant pages we found would point to other pages in that language or in other minority languages (Jauhiainen et al., 2019a). We, therefore, needed to do focused crawling and to give precedence to the links found on the pages written in the desired languages. As early as 2006, Boleda et al. (2006) were of the opinion that the technology was advanced enough to do large crawls in order to build web corpora. For such large web crawls, we also needed the crawler to be able to crawl for months if necessary. Obviously, the crawler needed to be polite and respect the general time limits for subsequent downloads from one server as well as the crawl limits and restrictions defined in the robots.txt files of the sites visited (Thelwall and Stuart, 2006; Emerson and O'Neil, 2006).

[11] https://www.basistech.com/text-analytics/rosette/language-identifier/

[12] https://nutch.apache.org

[13] https://github.com/saffsd/langid.py

3.1.2. Heritrix

As our web crawler, we chose Heritrix (Jauhiainen et al., 2015a; Jauhiainen et al., 2019a), a web archiving system developed by the Internet Archive (Mohr et al. (2004), see also Emerson and O'Neil (2006)).[14] Heritrix is used by several national libraries around the world to collect national web archives and it has been successfully used to collect text corpora by Baroni and Kilgariff (2006), Emerson and O'Neil (2006), Ferraresi et al. (2008), Baroni et al. (2009), Pomikálek et al. (2009), Schäfer and Bildhauer (2012) and Versley and Panchenko (2012). Heritrix obeys the robots.txt exclusion directives and has a system for giving precedence to specific links. Heritrix is open source and extendable, so we were also able to make custom changes to it.

3.1.3. Scope

When dealing with minority languages, the researchers usually have an idea which domains texts in the relevant languages could possibly be found in. We started collecting texts in Uralic minority languages by crawling, one by one, the .ee, .fi, .hu, .lv, .no, .ru, and .se domains (Jauhiainen et al., 2019a). According to Schäfer and Bildhauer (2013), large seed lists are only needed if one wants to find relevant pages as quickly as possible. Since we were crawling for minority pages anywhere in the national domains and hence were conducting large, long-lasting crawls, we seeded them with links to the home pages of the universities in these countries. We hoped that these sites with many outlinks would allow us to have very broad crawls in the long run. The university pages might also contain links from research projects to sites in small Uralic languages.

As we were building corpora from the texts found on the Internet, we were not interested in the links pointing to files not containing natural language, such as pictures (Jauhiainen et al., 2015a; Jauhiainen et al., 2019a). Such a strategy of ignoring the media files was also used for web corpus building by, for example, Baroni and Kilgariff (2006) and Ferraresi et al. (2008). After intensive testing with large crawls, we ended up using 600 threads at once, crawling only up to 20 links away from the seeds (in order to avoid, for example, getting stuck in a calendar or an online shop) and retiring download queues after they reached 100,000 links (as some sites may be replicating pages).

We conducted one-month long crawls for each of the national domains we were interested in. After one month of crawling, the number of queues was already quite low and the average speed had gone from about 300 URLs per second down to under 5 (under 100 in the .ru domain). Later, we conducted a two-month crawl which, in addition to all the relevant national domains, included the .com domain. This crawl was seeded with the URLs of the relevant pages found in the previous crawls.

3.2. Automatic Language Identification

3.2.1. Improving Language Identification

Part of the Suki project was dedicated to improving the state-of-the-art of language identification in texts and we further developed the language identification method by Jauhiainen (2010). Our language identifiers have fared very well in several shared tasks dedicated to distinguishing between close languages (Jauhiainen et al., 2018a; Jauhiainen et al., 2018b; Jauhiainen et al., 2019c). For collecting corpora in minority languages, we used an implementation of the HeLI method (Jauhiainen et al., 2016), based on which we created a language identifier service[15] that takes in text and responds with a corresponding language code. Currently the language identifier in production can distinguish between c. 400 languages and dialects in out-of-domain contexts (Jauhiainen et al., 2017). At the beginning of the project, we were able to find suitable training material for 34 of the 38 Uralic languages (Jauhiainen et al., 2015a; Jauhiainen et al., 2019a).[16] Hungarian, Finnish, and Estonian were not relevant as they are majority languages and thus we had 31 Uralic minority languages.

3.2.2. Language Identification While Crawling

In order to use language identification while crawling, we made some custom changes to the code of Heritrix. After downloading a file, we stripped the text of all the HTML markup before it was sent to the language identifier service. Our initial idea was to identify the text of the whole page, but in doing so we quickly encountered problems with speed. Identifying the whole page took too long when it was done while crawling. The crawler was able to process up to 400 pages per second and we needed the language identifier service to be able to keep up with that speed. We solved the problem by taking three excerpts of 100 characters from the entire text. Pages with fewer than 300 characters were ignored. The excerpts were sent as one package to the language identifier service where each was identified separately (Jauhiainen et al., 2015a; Jauhiainen et al., 2019a). If even one of the excerpts was identified as having been written in one of the languages of interest, the whole text of the page was sent to be identified. If the text of the entire page was still identified as one of the small Uralic languages, the text of the page was archived. The links found on such pages were given precedence over links from other pages in the frontier queue of the crawler (Jauhiainen et al., 2015a; Jauhiainen et al., 2019a).

3.3. Language Set Identification

The texts of web pages are often multilingual (Kilgarriff and Grefenstette, 2003), especially those including minority languages (Boleda et al., 2006). Most language identification methods are, however, built for identifying the language of a monolingual text (Lui et al., 2014; Jauhiainen et al., 2015b). When a monolingual language identifier is used to identify the language of a text written in multiple languages, it might, depending on the algorithm, produce an answer unrelated to the actual languages within the text (Prager, 1999). In the Suki project, we encountered this problem in practice when we were manually verifying the languages of the web pages automatically identified to be relevant. With language set identification, we refer to the

[14] http://www.archive.org

[15] https://github.com/tosaja/TunnistinPalveluFast

[16] No digitally encoded texts were found for Akkala, Ter, and Pite Saami languages nor for the Kamas language.

task of determining which languages are present in a document or text segment (Lui et al., 2014; Jauhiainen et al., 2015b).

3.3.1. The Method

Even though multilingual language identification for corpora creation purposes had been studied previously (Ludovik and Zacharski, 1999), there was no suitable off-the-shelf multilingual language identifier for us to use. For our project, we developed a new language set identification method (Jauhiainen et al., 2015b) which we named MultiLI.[17] The method uses a fixed size character window and, as the window slides stepwise along a text, the text of each window is identified with a language identifier. The language of the first window is stored in a variable called "current language" and when the language of subsequent windows has been different from the "current language" variable more times than a threshold, the language of the variable is changed. The method keeps track of all the languages that have been the "current language" at some point and returns these languages as a list.

3.3.2. Post-Crawl Language Set Identification

After the crawls of the national domains and the .com domain, all the texts found to possibly contain relevant languages were re-processed with MultiLI. Using the language set identifier, we could more easily find the pages containing any of the target languages (Jauhiainen et al., 2019a).

In addition to a list of languages, MultiLI provides the approximate percentages of those languages in the text of the whole page. As we did not want to miss any pages containing even a small amount of text in a relevant Uralic language, we accepted all texts of which at least 2% was in one of them.

One important function that MultiLI provided at this stage was the unknown language or "junk" detection. By not accepting any pages that were identified to have more than 9 languages, we did get rid of many pages that did not contain proper text at all.

We also downloaded the Common Crawl archive from December 2014.[18] We first used HeLI to identify the languages of the almost two billion pages in the archive. We then performed a more precise analysis with MultiLI on the 155,000 texts that had been identified by HeLI as having been written in a Uralic minority language. In this way, we found many new relevant links outside the national domains we had crawled ourselves (Jauhiainen et al., 2019a).

3.4. Crowdsourcing

There is a limit to how accurate automatic language identification can be. The accuracy of the identifier depends on the similarity of the training data to the texts that the crawler encounters in the wild. Even though the number of languages known by the language identifier can be high, it will almost certainly encounter languages it does not know. It will also encounter non-lingual or multilingual texts that can resemble one or more of the relevant languages (Schäfer and Bildhauer, 2013, 58). It is also possible that one of the relevant languages can be encountered which is written using a previously unknown orthography or is simply from a completely different domain than the material used for training, both of which prevent us from tightening the precision of the identifier too much.

As we ourselves were not familiar with most of the small Uralic languages, we planned to outsource the language verification to native speakers and linguists using crowdsourcing. One of the goals of the Suki project was to create a portal page with links to web pages that had been written in the relevant languages. We included the necessary crowdsourcing functionality into the portal site, which we call Wanca.[19] We were and are still not aware of similar platforms available for this purpose, which led us to develop our own.[20]

After removing exact duplicates, we uploaded to Wanca all the URLs of the texts that were still thought relevant after language set identification (Jauhiainen et al., 2019a). The URLs were accompanied by the language tag given by MultiLI as the most prominent relevant language for each page. In Wanca, registered users can vote for or against the language currently assigned to a page. The native speakers and linguists were advised to try to determine the largest relevant language for each page. A user with "expert user" rights is also allowed to verify the current language, which removes the voting option from the platform. An "expert user" could also change the current language to another, thus verifying the new language. The operations of verifying and changing the language could also be done to a whole website at once.

Originally we had published almost half a million links in Wanca. Since 2015, many automatically identified languages have been verified and, more importantly, almost 200,000 links that turned out not to be relevant have been discarded by us or the other users of the Wanca platform. By the time of the writing, Wanca contains 288,799 links that are considered to have been written in a Uralic minority language.

3.5. Sentence corpora pipeline

Since we do not automatically have copyrights for the downloaded texts (Fletcher, 2012; Schäfer, 2016), our aim was to create sentence corpora under the assumption that one sentence out of context can very rarely be considered to have individual copyright (Fletcher, 2012). We have described the sentence corpora creation pipeline in detail in an earlier article (Jauhiainen et al., 2019a).[21] In short, we started with the URLs tagged with relevant languages in Wanca and re-ran the corresponding texts through the language set identifier MultiLI. This time we carried the language set information forward and used it later to narrow

[17] https://github.com/tosaja/TunnistinPalveluMulti

[18] http://commoncrawl.org/2015/01/december-2014-crawl-archive-available/

[19] http://suki.ling.helsinki.fi/wanca/

[20] https://github.com/uhdigihum/WancaPlatform

[21] https://github.com/uhdigihum/SUKISentencePipeline

down the repertoire of languages available for the identifier. We divided the texts into individual lines keeping only those that our sentence tokeniser would later be able to find sentences from. We processed each line using the language set identifier and allowed the identifier only to indicate those relevant languages that were indicated in the set of the whole page. Each line was then split into sentences using a sentence tokenisation algorithm common to the languages involved (Jauhiainen et al., 2019a) using abbreviation guessing heuristics presented by (Mikheev, 2002). Afterwards, the language of each individual sentence was identified using MultiLI and the most prominent language indicated by it was set as the language of the sentence. We removed duplicate sentences as the Internet is full of web services that automatically generate text in natural languages, and the duplicate sentences probably do not represent any natural frequency used by humans. Finally, sentences written in relevant languages were added to corresponding sentence collections.

3.6. Corpora

From time to time, we have re-crawled the links that are considered relevant to see if the pages still exist. Since we first crawled for texts in Uralic minority languages, 80% of the links and 90% of the sites we found have either disappeared or their robots.txt directives have been tightened. To create the sentence corpora, we used a re-crawl from 2016 where we had texts from 119,052 pages. After our pipeline, we ended up with 646,043 unique sentences in 29 languages. The sentences come from 39,731 pages. The sentence corpora created in the Suki project were published at the Language Bank of Finland in their Korp service in 2019.[22] A downloadable version of the corpora was made available in the spring of 2020.[23]

4. Lessons learned

We only created our sentence corpora pipeline after the languages of the pages had been curated in Wanca by us and the language experts. The amount of junk and texts in unknown languages was considerable. In the beginning, we discarded complete sites as junk as the sources for language identification errors were apparent after inspecting only a few pages within a site. Many of the errors were in the parts of the texts which did not contain complete sentences, but were, for example, lists of the names of mechanical components. This is why we suggest using the sentence creation pipeline even before uploading the URLs to a crowdsourcing platform. Only those pages where at least one proper sentence is written in one of the relevant languages should be forwarded to manual inspection. This would probably get rid of much junk and irrelevant pages without anyone having to go through them manually. It is important that the native speakers and linguists donating their time and skills feel that their work is valuable and meaningful.

The parameters of the language identifier must be adjusted so that it is able to keep up with the speed of the crawler. When the HeLI method uses a very large word and character *n*-gram vocabulary it is slow to start. However, when it is offered as a service and the models are already loaded in memory, the size of the models does not essentially affect the speed of the identification process. The number of languages available to the identifier, however, has an impact on the identification speed linear to the number of languages.

The number and size of the excerpts sent to the language identifier while crawling must be decided, taking into account the capabilities of the hardware used. We used three short excerpts, but if the hardware allows, more excerpts can be tested. If the relevant languages can be identified with sufficient recall using shorter segments, then more of these shorter excerpts could be used. Thus, the length of each excerpt is determined by the accuracy of the language identifier used. As a shorter segment of text is less likely to contain several languages, using shorter excerpts also helps in dealing with multilingual pages.

When one is targeting minority languages, one does not want to miss any potential texts while crawling, hence errors in precision are much more acceptable than errors in recall. In hindsight, if the three excerpts contained a relevant language, re-identifying the language of the whole text while crawling was a mistake. The later stages of our process would have removed the incorrectly identified texts, but since we relied on the identification of the language of the whole text, we may have missed some multilingual ones.

The HeLI method is very fast and precise when dealing with languages that can be separated into words easily by, for example, using whitespaces as delimiters. In case the relevant languages include ones that do not use whitespaces, we suggest employing other methods. For example, we used Naive Bayes in our winning submission to the track for traditional Chinese of the Discriminating between the Mainland and Taiwan variation of Mandarin Chinese (DMT) shared task (Jauhiainen et al., 2019c).

One reason why we have not been eager to publish the language models of the service in production together with our code is the fact that we have only been improving the recall and precision of the languages relevant to the Suki project. As the source texts for other language models were mostly gathered from Wikipedia, some of these models, for example English, have severe problems with recall and precision. Another reason is that the complete models used in production with the crawler take in total 20 gigabytes of space and it is not trivial to distribute files of this size. It is future work to prune or optimise the data structure without losing identification speed or accuracy.

5. Proposed strategy

In this section, we introduce the strategy that we recommend for web sentence corpora creation for minority languages based on our trials and documented experience. First, we present the suggested stages of the web corpora building strategy. Second, we list the technical components that can be used to implement the strategy.

5.1. Strategy outline

Stage 1. Decide which top-level Internet domains are relevant to your crawl. Gather a list of prominent websites for each top-level domain to use as seeds for each crawl.

[22] http://urn.fi/urn:nbn:fi:lb-2019052401

[23] http://urn.fi/urn:nbn:fi:lb-2020022901

If you already know of websites written in the language of interest, use them as seeds as well.

Stage 2. Start a breadth-first crawl within the given domain and identify the possible languages of each page by taking as many extracts from the page as is possible given the speed of your language identifier service. Use a language identifier optimised for recall of the relevant languages, as you do not want to miss any possible sources at this point. Precision for the relevant languages can be sacrificed if more speed is needed. If even one extract is indicated to be written in a relevant language, store the whole text.

Stage 3. After the crawl, remove duplicate or near-duplicate texts. We suggest removing duplicates that differ from each other only by non-alphabetic (or non-logographic) characters, for example by different time-stamps.

Stage 4. Process all the stored texts using a language set identifier. If too many languages are detected for one file it is probably written in a language unknown to the language identifier or contains large amounts of non-lingual material, such as lists of product codes. If the set of identified languages is reasonable, keep texts that include at least one language relevant to your corpus. Store the language set and the URL as text-specific metadata.

Stage 5. Segment the texts into sentences and retain only complete sentences. If your sentence tokeniser cannot span line-breaks, you can first use the language set identifier to identify each line and keep only those in the relevant languages.

Stage 6. Identify the language of each sentence using only the relevant languages indicated by the previous level language set identification. Tag each sentence with the majority language indicated by the language set identifier.

Stage 7. Retain only those texts that include at least one sentence in a relevant language. Tag the retained texts with the relevant language in which the most sentences are written.

Stage 8. Use experts and native speakers to verify that the retained pages actually include relevant languages. In case some of the pages have been removed from the Internet, but the text had duplicates or near-duplicates, use the first working address from the duplicate list. Remove those texts that are clearly rejected by crowdsourcing.

Stage 9. In case you have special language-dependent sentence tokenisers for the relevant languages, you might want to re-tokenise the original text at this stage and then remove duplicate sentences within the same language. Otherwise, use the sentences and their identifications generated at stages 5 and 6.

Stage 10. Shuffle the sentences and add them to their language-specific collections.

5.2. Technical components

Web crawling To collect the texts for our "Wanca in Korp 2016" corpus (Jauhiainen et al., 2019b), we used Heritrix version 3.1. Currently, we have an operational version of Heritrix 3.3. The modified version of Heritrix 3.3 containing the enhanced text pre-processing and the ability to use a language identifier service is available on GitHub.[24]

Language identification We have published the monolingual language identifier service implementing the HeLI algorithm on GitHub.[25] While preparing this article, we improved the documentation and included character *n*-gram models from one to six characters as well as individual words for Finnish and Swedish as an example.

Language set identification The language set identifier MultiLI, which uses HeLI together with the language set identification algorithm is also available on GitHub.[26] It does not contain any language models, but it can use the same language models as the monolingual language identifier service.

Crowdsourcing platform The code for the Wanca platform is also available on GitHub.[27]

Sentence tokeniser The sentence tokeniser and the scripts for the whole sentence corpora pipeline are available on GitHub.[28]

6. Conclusions

We have presented the strategy we used to create sentence corpora for Uralic minority languages, analysed its usability, and suggested an improved version of the strategy. We believe that the strategy could be used to build web corpora for other minority languages that have web pages in the same national top-level domain as the majority language of the country or countries in which the languages are used. Building web corpora for minority languages is an important undertaking for the preservation of these under-resourced and often endangered languages. The crowdsourcing platform can be used to inform the native language users of the resources available online.

7. Acknowledgements

This work has been funded by the Kone Foundation, the Academy of Finland, and FIN-CLARIN. The authors would like to thank all those users of the Wanca platform who have contributed to the language verifying effort as well as the anonymous reviewers of this article for their valuable comments.

8. Bibliographical References

Aggarwal, C. C., Al-Garawi, F., and Yu, P. S. (2001). Intelligent Crawling on the World Wide Web with Arbitrary Predicates. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, pages 96–105, New York, NY, USA. ACM.

Arkhangelskiy, T. (2019). Corpora of social media in minority Uralic languages. In *of the Fifth International Workshop on Computational Linguistics for Uralic Languages (IWCLUL 2019)*, pages 125–140, Tartu, Estonia.

[24] `https://github.com/uhdigihum/heritrix3`

[25] `https://github.com/tosaja/TunnistinPalveluFast`

[26] `https://github.com/tosaja/TunnistinPalveluMulti`

[27] `https://github.com/uhdigihum/WancaPlatform`

[28] `https://github.com/uhdigihum/SUKISentencePipeline`

Barbaresi, A. (2013). Crawling microblogging services to gather language-classified URLs: Workflow and case study. In *Proceedings of the Student Research Workshop (ACL 2013)*, pages 9–15, Sofia, Bulgaria.

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

Baroni, M. and Bernardini, S. (2004). BootCaT: Bootstrapping Corpora and Terms from the Web. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal.

Baroni, M. and Kilgariff, A. (2006). Large linguistically-processed Web corpora for multiple languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 87–90, Trento, Italy.

Baroni, M. and Ueyama, M. (2004). Retrieving Japanese specialized terms and corpora from the World Wide Web. In *Proceedings of the 7th Conference on Natural Language Processing (KONVENS 2004)*, Wien, Austria.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Baykan, E., Henzinger, M., and Weber, I. (2008). Web Page Language Identification Based on URLs. In *Proceedings of the 34th International Conference on Very Large Data Bases (VLDB '08)*, pages 176–187, Auckland, New Zealand.

Biemann, C., Bildhauer, F., Evert, S., Goldhahn, D., Quasthoff, U., Schäfer, R., Simon, J., Swiezinski, L., and Zesch, T. (2013). Scalable Construction of High-Quality Web Corpora. *JLCL*, 28:23–59.

Boleda, G., Bott, S., Meza, R., Castillo, C., Badia, T., and Lopez, V. (2006). CUCWeb: a Catalan corpus built from the Web. In *Proceedings of the 2nd International Workshop on Web as Corpus (WAC '06)*, pages 19–26, Trento, Italy.

Chakrabarti, S., Van den Berg, M., and Dom, B. (1999). Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11):1623–1640.

De Bra, P., Houben, G.-J., Kornatzky, Y., and Post, R. (1994). Information Retrieval in Distributed Hypertexts. In *Proceeding of Computer-Assisted Information Retrieval (RIAO 1994)*, pages 481–491, NY, NY.

Diligenti, M., Coetzee, F. M., Lawrence, S., Giles, C. L., and Gori, M. (2000). Focused crawling Using Context Graphs. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB 2000)*, pages 527–534, Cairo, Egypt.

Emerson, T. and O'Neil, J. (2006). Experience Building a Large Corpus for Chinese Lexicon Construction. In Marco Baroni et al., editors, *WaCky! Working papers on the Web as Corpus*, pages 41–62. Bologna: GEDIT.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th International Workshop on Web as Corpus (WAC-4)*, pages 47–54, Marrakech, Morocco.

Fletcher, W. H. (2012). Corpus Analysis of the World Wide Web. In C.A. Chapelle, editor, *The Encyclopedia of Applied Linguistics*. American Cancer Society.

Ghani, R., Jones, R., and Mladenič, D. (2001). Mining the Web to Create Minority Language Corpora. In *Proceedings of the 10th International Conference on Information and Knowledge Management (ACM CIKM 2001)*, pages 279–286, Atlanta, Georgia.

Habernal, I., Zayed, O., and Gurevych, I. (2016). C4Corpus: Multilingual Web-size Corpus with Free License. In Nicoletta Calzolari, et al., editors, *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France.

Jauhiainen, H., Jauhiainen, T., and Lindén, K. (2015a). The Finno-Ugric Languages and The Internet Project. In *Proceedings of the 1st International Workshop on Computational Linguistics for Uralic Languages (IWCLUL 2015)*, number 2 in Septentrio Conference Series, pages 87–98.

Jauhiainen, T., Lindén, K., and Jauhiainen, H. (2015b). Language Set Identification in Noisy Synthetic Multilingual Documents. In A. Gelbukh, editor, *Proceedings of the Computational Linguistics and Intelligent Text Processing 16th International Conference, (CICLing 2015)*, Part I of Lecture Notes in Computer Science, pages 633–643, Cairo, Egypt. Springer.

Jauhiainen, T., Lindén, K., and Jauhiainen, H. (2016). HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2016)*, pages 153–162, Osaka, Japan.

Jauhiainen, T., Lindén, K., and Jauhiainen, H. (2017). Evaluation of Language Identification Methods Using 285 Languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa 2017)*, pages 183–191, Gothenburg, Sweden.

Jauhiainen, T., Jauhiainen, H., and Lindén, K. (2018a). HeLI-based experiments in Swiss German dialect identification. In *Proceedings of the 5th Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 254–262, Santa Fe, New Mexico.

Jauhiainen, T., Jauhiainen, H., and Lindén, K. (2018b). Iterative Language Model Adaptation for Indo-Aryan Language Identification. In *Proceedings of the 5th Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 66–75, Santa Fe, New Mexico.

Jauhiainen, H., Jauhiainen, T., and Linden, K. (2019a). Wanca in Korp: Text corpora for underresourced Uralic languages. In Jarmo Harri Jantunen, et al., editors, *Proceedings of the Research data and humanities (RDHUM) 2019 conference*, number 17 in Studia Humaniora Ouluensia, pages 21–40, Finland. University of Oulu.

Jauhiainen, H., Jauhiainen, T., and Lindén, K. (2019b). Wanca 2016, Korp Version (BETA). [text corpus]. Kielipankki. Retrieved from http://urn.fi/urn:nbn:fi:lb-2019052401.

Jauhiainen, T., Jauhiainen, H., and Lindén, K. (2019c).

Discriminating between Mandarin Chinese and Swiss-German varieties using adaptive language models. In *Proceedings of the 6th Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2019)*, pages 178–187, Minneapolis, Minnesota.

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., and Lindén, K. (2019d). Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Jauhiainen, T. (2010). Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki.

Kanerva, J., Luotolahti, J., Laippala, V., and Ginter, F. (2014). Syntactic N-gram Collection from a Large-Scale Corpus of Internet Finnish. In Andrius Utka, et al., editors, *Proceedings of the Sixth International Conference Baltic HLT 2014*, volume 268 of *Frontiers in Artificial Intelligence and Applications*, pages 184–191.

Kilgarriff, A. and Grefenstette, G. (2003). Web as Corpus. *Computational Linguistics*, 29(3):333–347.

Kornai, A., Halácsy, P., Nagy, V., Oravecz, C., Trón, V., and Varga, D. (2006). Web-based frequency dictionaries for medium density languages. In *Proceedings of the 2nd International Workshop on Web as Corpus (WAC '06)*, Trento, Italy.

Kuznetsova, N., Markus, E., and Muslimov, M. (2015). Finnic Minorities of Ingria. In Heiko F. Marten, et al., editors, *Cultural and Linguistic Minorities in the Russian Federation and the European Union: Comparative Studies on Equality and Diversity*, pages 127–167. Springer.

Ludovik, Y. and Zacharski, R. (1999). Multilingual Document Language Recognition for Creating Corpora. Technical report, New Mexico State University.

Lui, M., Lau, J. H., and Baldwin, T. (2014). Automatic Detection and Language Identification of Multilingual Documents. *Transactions of the Association for Computational Linguistics*, 2:27–40.

Lyding, V., Stemle, E., Borghetti, C., Brunello, M., Castagnoli, S., Dell'Orletta, F., Dittmann, H., Lenci, A., and Pirrelli, V. (2014). The PAISÀ Corpus of Italian Web Texts. In *Proceedings of the 9th International Workshop on Web as Corpus (WaC-9)*, pages 36–43, Gothenburg, Sweden.

Medelyan, O., Schulz, S., Paetzold, J., Poprat, M., and Marcó, K. (2006). Language Specific and Topic Focused Web Crawling. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, pages 865–868, Genoa, Italy.

Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods. In *Proceedings of the 14th International Conference on Machine Learning*, pages 227–235.

Mikheev, A. (2002). Periods, Capitalized Words, etc. *Computational Linguistics*, 28(13):289–318.

Mohr, G., Stack, M., Rnitovic, I., Avery, D., and Kimpton, M. (2004). Introduction to Heritrix. 4th International Web Archiving Workshop (at ECDL2004).

Mon, P. Y. and Mikami, Y. (2011). Myanmar Language Search Engine. *International Journal of Computer Science Issues (IJCSI)*, 8(2):118–126.

Murphy, B. and Stemle, E. W. (2011). PaddyWaC: A minimally-supervised web-corpus of hiberno-English. In *Proceedings of the 1st Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 22–29, Edinburgh, Scotland. Association for Computational Linguistics.

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

Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S. P., and Biemann, C. (2018). Building a Web-Scale Dependency-Parsed Corpus from CommonCrawl. In Nicoletta Calzolari, et al., editors, *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Pomikálek, J., Rychlý, P., and Kilgarriff, A. (2009). Scaling to Billion-plus Word Corpora. *Advances in Computational Linguistics*, 41(3):3–13.

Pomikálek, J., Jakubícek, M., and Rychlỳ, P. (2012). Building a 70 billion word corpus of English from ClueWeb. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 502–506, Istanbul, Turkey.

Prager, J. M. (1999). Linguini: Language Identification for Multilingual Documents. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences (HICSS-32)*, Maui, Hawaii.

Priyatam, P. N., Vaddepally, S., and Varma, V. (2012). Domain Specific Search in Indian Languages. In *Proceedings of the 1st workshop on Information and knowledge management for developing regions (IKM4DR'12)*, pages 23–30, Maui, Hawaii.

Rehůřek, R. and Kolkus, M. (2009). Language Identification on the Web: Extending the Dictionary Method. In A. Gelbukh, editor, *Proceedings of the Computational Linguistics and Intelligent Text Processing 10th International Conference, (CICLing 2009)*, Lecture Notes in Computer Science, pages 357–368, Mexico City, Mexico. Springer.

Scannell, K. P. (2007). The Crúbadán Project: Corpus building for under-resourced languages. *Cahiers du Cental*, 5(1):1–10.

Schulz, S., Lyding, V., and Nicolas, L. (2013). Compiling a diverse web corpus for South Tyrolean German - STirWaC. In Stefan Evert, et al., editors, *Proceedings of the 8th Web as Corpus Workshop (WAC-8)*, pages 37–45.

Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Schäfer, R. and Bildhauer, F. (2013). *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, San Francisco.

Schäfer, R., Barbaresi, A., and Bildhauer, F. (2014). Focused Web Corpus Crawling. In *Proceedings of the 9th International Workshop on Web as Corpus (WaC-9)*, pages 9–15, Gothenburg, Sweden.

Schäfer, R. (2016). CommonCOW: Massively Huge Web Corpora from CommonCrawl Data and a Method to Distribute them Freely under Restrictive EU Copyright Laws. In Nicoletta Calzolari, et al., editors, *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France.

Sharoff, S. (2006a). Open-source Corpora: using the net to fish for linguistic data. *International Journal of Corpus Linguistics*, 11(4):435–462.

Sharoff, S. (2006b). Creating general-purpose corpora using automated search engine queries. In Marco Baroni et al., editors, *WaCky! Working papers on the Web as Corpus*, pages 63–98. Bologna: GEDIT.

SIL. (2013). *ISO 639-3 Codes for the representation of names of languages*. SIL International.

Gary F. Simons et al., editors. (2018). *Ethnologue: Languages of the World, Twenty-first edition*. SIL International, Dallas, Texas.

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., and Lopez, A. (2013). Dirt Cheap Web-Scale Parallel Text from the Common Crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria.

Somboonviwat, K., Tamura, T., and Kitsuregawa, M. (2005). Simulation Study of Language Specific Web Crawling. In *Proceedings of the 21st International Conference on Data Engineering Workshops (ICDEW'05)*, Tokyo, Japan.

Spoustová, J. and Spousta, M. (2012). A High-Quality Web Corpus of Czech. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pages 311–315, Istanbul, Turkey.

Suchomel, V. and Pomikálek, J. (2012). Efficient Web Crawling for Large Text Corpora. In *Proceedings of the 7th International Workshop on Web as Corpus (WAC-7)*, pages 39–43, Lyon, France.

Tamura, T., Somboonviwat, K., and Kitsuregawa, M. (2007). A Method for Language-Specific Web Crawling and Its Evaluation. *Systems and Computers in Japan*, 38(2):10–20.

Thelwall, M. and Stuart, D. (2006). Web Crawling Ethics Revisited: Cost, Privacy, and Denial of Service. *Journal of the American Society for Information Science and Technology*, 57(13):1771–1779.

Ueyama, M. and Baroni, M. (2005). Automated construction and evaluation of Japanese Web-based reference corpora. In *Proceedings of 3th Corpus Linguistics Conference*, Birmingham, United Kingdom.

Versley, Y. and Panchenko, Y. (2012). Not Just Bigger: Towards Better-Quality Web Corpora. In *Proceedings of the 7th International Workshop on Web as Corpus (WAC-7)*, pages 45–52.

Wagner Filho, J. A., Wilkens, R., Idiart, M., and Villavicencio, A. (2018). The brWaC Corpus: A New Open Resource for Brazilian Portuguese. In Nicoletta Calzolari, et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.