hrWaC and slWac: Compiling Web Corpora for Croatian and Slovene

Nikola Ljubešić[1] and Tomaž Erjavec[2]

[1] Faculty of Humanities and Social Sciences, University of Zagreb, Croatia
nikola.ljubesic@ffzg.hr
[2] Dept. of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Abstract. Web corpora have become an attractive source of linguistic content, yet are for many languages still not available. This paper introduces two new annotated web corpora: the Croatian hrWaC and the Slovene slWaC. Both were built using a modified standard "Web as Corpus" pipeline having in mind the limited amount of available web data. The modifications are described in the paper, focusing on the content extraction from HTML pages, which combines high precision of extracted language content with a decent recall. The paper also investigates text-types of the acquired corpora using topic modeling, comparing the two corpora among themselves and with ukWaC.

Keywords: web corpus, Croatian, Slovene, topic modeling.

1 Introduction

With the advent of the web, a vast new source of linguistic information has emerged. The exploitation of this resource has especially gained momentum with the WaCky initiative [1], which has popularised the concept of "Web as Corpus". It has also made available tools for compiling such corpora and produced large WaC corpora for a number of major European languages. Now such corpora are also being built for the so called smaller languages, such as Norwegian [2] and Czech [3], moving the concept of a "large corpus" for smaller languages up to the 1 billion token frontier. As Web corpus acquisition is much less controlled than that for traditional corpora, the necessity of analyzing their content gains in significance. The linguistic quality of the content is mostly explored through word lists and collocates [1] while the content itself is explored using unsupervised methods, such as clustering and topic modeling [4].

2 Building the hrWaC and slWaC

The standard pipeline for building web corpora was developed primarily for languages where the amount of web data is orders of magnitude larger than the corpus being built. On the other hand, smaller languages cannot afford the luxury of sampling since the amount of data is limited - alternatively, this can be seen as a bonus, as a large

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 395–402, 2011.

portion of all the language Web can be turned into a language corpus.[1]. In this paper we propose a modified traditional pipeline which would better suit smaller languages with a limited amount of web data available. Additionally, we describe a novel content extraction method with high precision and a decent recall.

A detailed overview of the numerical results of applying the pipeline for the two corpora is given in Table 1.

Table 1. Numerical summary of the corpus creation process. This version of the corpora was crawled between January and March, 2011.

	hrWaC	slWaC
# of seed domains	12,033	11,493
# of domains crawled	16,398	18,418
# of crawled documents	15,747,585	9,247,341
# of documents after deduplication	14,654,394	9,022,716
# of extracted documents	3,924,194	1,598,011
# of language identified documents	3,607,054	1,337,286
# of non-filtered documents	3,409,226	1,287,895
# of tokens	1,186,795,086	380,299,844

2.1 Collecting Seeds, Crawling, Physical Deduplication

For collecting seed URLs we used the Yahoo! Search BOSS API. The Yahoo search index was queried with random bigrams composed of mid-frequency tokens (frequency rank 1,000-10,000 from 100-million-token newspaper corpora) and about 50,000 URLs were collected for every language. Since Croatian and Slovene webs are much smaller than those for "large" languages an early decision was made not to sample the web as in the case of English, German or French, but to crawl it completely. For that reason only top pages of the collected domains on the *hr* and *si* top domains were used as seeds for the crawling process.

Crawling was performed with a multi-threaded, breadth-first crawler developed for this purpose since most available crawlers lack precise control, such as filtering by the MIME file type. Only "text/html" files of size between 50 and 500 kB were crawled. Both crawls ran several weeks and collected 15.7 million Croatian and 9.2 million Slovene documents.

The next step in the pipeline was physical deduplication, i.e. removing all but one copy of files that are physically identical. For that task the SHA224 hashing algorithm was used. During the process 2.25% of the Croatian documents and 2.3% of the Slovene ones were removed.

[1] Estimating the number of documents per language via Google (five most frequent words with language filter) on 2011-05-27 yields these numbers: English 25.27 billion (4.7 billion on *uk* domain), German 2.27 billion (1.27 billion on *de* domain), Norwegian 332 million (261 million on *no* domain), Croatian 229 million (70 million on *hr* domain), Slovene 210 million (82 million on *si* domain).

2.2 Content Extraction

A crucial step in building a web corpus is the content extraction step, often called boilerplate removal. We prefer the first term since it is our belief that just a part of the HTML document should be retained, rather than just a portion of the document removed.

This processing step has undergone the most changes in contrast to the classic WaCky pipeline. We did not use the well known body text extraction (BTE) algorithm, but a novel, more conservative algorithm, which aims at a very high precision, but without too great penalties on the recall. In our opinion this is the phase where most noise enters the corpus, which can have negative impact not only on the linguistic quality of the corpus, but also indirectly on its size. An example of boilerplate removal coupled with near-duplicate removal possibly gone bad is the Norwegian web corpus noWaC [2] where on the near-duplicate removal step 90% of documents were removed, probably because of boilerplate remains, which then identified many documents as near-duplicates.

Our algorithm is based on the notion that the largest amount of linguistically relevant content can be found by identifying the largest chunk of graphically identical and linguistically correct formatted text in the document. Since almost all web sites nowadays use CSS for styling and define the CSS class used in *id* and *class* attributes of the HTML elements, it was our assumption when building the algorithm that by identifying the largest amount of text on the same depth in the DOM node tree with same formatting we will identify the document body. Therefore our rather simple algorithm proceeds as follows:

1. Pre-format the HTML document by enclosing all text divided by *br* elements into separate paragraph elements
2. Represent every paragraph node in the DOM node tree as the path of triples (*tag*, *id attribute*, *class attribute*) from the root node down to the node of interest[2]
3. For every paragraph element which satisfies the constraints of well formatted paragraphs defined by a series of regular expressions calculate its weight via the formula

$$weigth(p) = \frac{text\_length(p) * (1 - link\_density(p))}{size(map_{tic:w})} \quad (1)$$

   where the $text\_length()$ function returns the number of characters in the paragraph, the $link\_density()$ function returns the percentage of characters being part of a link and the $size(map_{tic:w})$ expression returns the number of elements in the map where the sum of weights of elements with specific tag-id-class paths is stored.
4. Add the calculated weight to the $map_{tic:w}$ map under the corresponding tag-id-class path.
5. Choose the tag-id-class path with the maximum weight and return the textual content of all previously analyzed elements having that tag-id-class path.

By multiplying the text length with the percentage of the not-linked text we take into account only the amount of "clean" text while by dividing it further with the number

[2] An actual example of such path is *((div, container,), (div, wrap,), (div, content, content_article), (div, article_text, article_text))*. Only the nodes below the *body* node are recorded since higher nodes are constant.

of different tag-id-class paths found up to that moment, larger weights are given to the elements found sooner while traversing the DOM tree. The weighting function coupled with the constraints of well formatted paragraphs follows the intuition that the main text of a document will be the largest amount of linguistically well formatted text not containing many links which is found rather at the beginning of the document. Same graphical formatting is ensured by the uniformity of the tag-id-class path.

An evaluation of the algorithm was performed on 200 online newspaper documents downloaded from 20 different news portals.[3] As competing methods we chose the BTE algorithm, due to its heavy usage in the WaCky community as implemented in BootCaT[4] and the BoilerPipe 1.1.0 API,[5] due to its recent popularity in the NLP/IR community. In the experiment we call our algorithm ContentExtractor. The precision and recall evaluation measures were calculated via LCS (longest common subsequence) where the result was normalized for precision by the length of the extracted text while for recall the result was normalized by the length of the gold standard. The results of the experiment are given in Table 2.

Table 2. Precision, recall and F1 of three different algorithms on the task of content extraction

	precision	recall	F1
ContentExtractor	0.979	0.707	0.821
BTE	0.570	0.955	0.713
BoilerPipe	0.847	0.921	0.882

The results show an best overall performance of the BoilerPipe algorithm. On the other hand, BTE has an even greater recall than BoilerPipe but with a drastic drop in precision. The distinction of ContentExtractor is a very high precision, but with lower recall. Obviously, each of the algorithms has its advantages. If one was aiming at high recall, being ready to clean up the result in later stages, BTE could be a good solution. A downside of the implementation of the BTE algorithm in BootCat is that it looses all structural information, and therefore makes latter clean-up very complicated. On the other hand, the other two algorithms keep the paragraph structure intact. If one needed a middle approach with both decent precision and recall, the BoilerPipe algorithm would be the best choice. It is our belief that collecting HTML data primarily for the purpose of modeling linguistic phenomena, loosing some text elements like titles, headings and lists is a tolerable (if not desirable?) loss. The omission of these text structures enables very high precision; also, today's corpus investigations seldom cross paragraph boundaries. One could wonder on this stage why we chose an algorithm with lower recall having in mind the smaller amount of available data in the first place. It is our belief

[3] An implementation of the algorithm and the evaluation sample are published on `http://www.nljubesic.net/hrWaC_Croatian_Web_Corpus/content_extraction.html`

[4] The PotaModule module was obtained from `http://bootcat.sslmit.unibo.it/`

[5] The ArticleExtractor class optimized for newspaper articles was used `http://code.google.com/p/boilerpipe/`

that this is an inevitable data loss if one wanted to obtain a clean and thereby usable resource.

Since we made an early decision to avoid the step of near duplicate removal because of (I) the danger of loosing a large amount of data on false positives (II) our belief that the problem of repeating content on smaller webs is much smaller than on the webs of larger languages and (III) its overall complexity, we bundled an additional step of duplicate removal with content extraction by removing identical paragraphs on the level of each domain.

By using the developed algorithm to extract text from the crawled documents for hrWaC and slWaC the conservativeness of our approach is shown by the fact that ContentExtractor returned text from only 26.5% of Croatian and 17.8% of the Slovene documents.

2.3 Language Identification, Filtering and PoS Tagging

After extracting linguistic content from HTML documents, we performed language identification with a combination of a second-order Markov chain model and a function word filter for 22 languages. We lowered the level of language identification to the paragraph level since our research showed that the error rate on paragraph level by combining two classifiers in a smart fashion is the same as with second-order Markov chain models on the document level [6]. Through the language identification step we experienced an approximately 8% document loss in Croatian and 17% document loss in Slovene. The higher loss in the Slovene corpus could be due to its membership in the European union and the consequent larger number of documents on the Slovene domain written primarily in English.

Additional filtering was performed to eliminate too short documents, those with encoding errors and those with a high amount of punctuation (often no running text like lists, document abstracts etc.) At the document filtering step 5.4% of Croatian and 3.7% of Slovene documents were removed.

The Croatian corpus was PoS-tagged and lemmatised with the tagger developed in the Institute for Linguistics at the Faculty of Humanities and Social Sciences, University of Zagreb [7], while the Slovene corpus was tagged and lemmatised with ToTaLe [8] trained on JOS corpus data [9]. The two taggers share harmonised PoS or, better, MSD (morphosyntactic description) tagsets, as both follow the MULTEXT-East morphosyntactic specifications [10].

As shown in Table 1, the final number of tokens is 1.2 billion for hrWaC and 380 million for slWaC. However, this is just a first version of the two corpora and our intention is to continue collecting new data with the primary goal of enhancing the size of the Slovene corpus.

3 Corpus Comparison

In this section we explore the content of the web corpora through the topic modeling method already used for corpus analysis tasks [4]. Our models were built with MALLET, [11] used with the default settings.

Table 3. Twenty hrWaC and slWaC topics with the amount of text they cover and up to ten words with highest probability. Topic names in bold are present in both hrWaC and slWaC.

Lg	Topic name	Size	Words with highest probability
sl	intl. politics	4.7%	leto država vojna človek predsednik oblast zda napad vojska dan
hr	**reg. politics**	5.9%	zemlja srbija predsjednik godina država rat vlada hrvatska
sl	**reg. politics**	3.6%	država slovenija eu leto članica minister predsednik hrvaška
hr	**dom. politics**	6.2%	predsjednik vlada stranka izbor sanader ministar pitanje zakon
sl	**dom. politics**	4.9%	vlada predsednik zakon stranka slovenija minister sodišče
hr	**law**	3.0%	zakon tema odluka pravo postupak sud članak osoba ugovor
sl	**law**	4.4%	zakon podatek primer pravica člen oseba plačilo storitev dan k
hr	crime	5.3%	policija sud godina osoba slučaj zatvor kazna sat policajac sudac
hr	**finance**	7.1%	godina kuna milijun tvrtka cijena banka euro tržište dionica
sl	**finance**	5.2%	leto evro odstotek podjetje milijon družba banka cena trg država
hr	**sports**	2.0%	utrka mjesto godina natjecanje prvenstvo vrijeme sezona staza
sl	**sports**	4.0%	tekma minuta igra leto točka prvenstvo igralec ekipa mesto
hr	soccer	4.8%	utakmica igrač klub momčad minuta pobjeda liga sezona trener
sl	classified ads	2.9%	oglas iskanje seznam stran znamka stroj možnost vrh kvadrat
sl	environment	6.0%	voda energija prostor barva material sistem uporaba del naprava
hr	**automoto**	3.8%	motor automobil vozilo model boja auto sustav dio energija
sl	**automoto**	2.6%	vozilo avtomobil motor dirka vožnja leto voznik avto kolo mesto
hr	web	4.8%	dan godina stranica članak web informacija rubrika hr internet
hr	**IT**	4.2%	korisnik internet uredaj igra stranica slika računalo program
sl	**IT**	6.1%	stran uporabnik podatek sistem računalnik slika program
hr	construction	5.5%	grad područje cesta dio voda godina stan kuća zgrada prostor
hr	**local themes**	7.0%	godina grad županija dan škola sat udruga zagreb izložba rad
sl	**local themes**	5.4%	občina leto cesta mesto prostor ljubljana članek območje
hr	**education**	6.4%	rad godina projekt program škola razvoj sustav područje student
sl	**education**	8.0%	delo področje program projekt leto šola razvoj organizacija
hr	**health**	4.2%	bolest dan hrana voda godina koža lijek liječnik tijelo ulje
sl	**health**	3.6%	telo bolezen zdravilo zdravnik leto otrok bolnik zdravljenje
hr	**travel**	3.8%	hotel brod sat more mjesto otok grad godina dan soba
sl	**travel**	5.3%	mesto dan pot ura leto hotel morje otok čas soba
hr	**family**	9.3%	čovjek dan vrijeme mi život žena put djeca stvar godina
sl	**family**	10.8%	človek čas življenje stvar svet ženska otrok dan način moški
hr	**religion**	3.7%	čovjek crkva život bog svijet knjiga vrijeme godina riječ djelo
sl	**religion**	3.2%	otrok leto cerkev dan oče bog človek čas mati roka
hr	forum	3.1%	mi čovjek stvar dan vec jel problem par kajati godina
sl	lifestyle	4.0%	koža hrana voda olje žival pes rastlina izdelek vrsta las
sl	art	4.4%	leto knjiga delo razstava ljubljana avtor umetnost jezik zbirka
hr	**movies**	6.2%	film godina svijet priča uloga glumica žena glumac serija život
sl	**movies**	5.5%	film leto vloga igralec režiser zgodba nagrada svet igralka serija
hr	**music**	3.7%	pjesma album koncert godina glazba festival publika predstava
sl	**music**	5.4%	leto skupina glasba pesem koncert festival album dan oddaja

The documents used for training the topic models were a 10% random sample of the corpus documents with their content stripped down to noun lemmata. Experimenting with the amount of data necessary for constant resuts showed that modeling on 1/10 of randomly chosen data does not change the results significantly, but, of course, does speed up this computationally demanding task significantly.

The number of the topics was set to 20, and the topics were manually named by examining the topic keywords. The resulting topics are shown with their size, counted as the number of tokens, and up to ten most probable terms in Table 3 for the two corpora. As can be seen, the two topic modeling results are quite similar. Fifteen out of twenty topics can be considered almost identical. The more prominent topics on the Croatian side are crime, soccer, the web, construction and on-line forums, while the topics prominent for Slovene are international politics, classified ads, environment, lifestyle and art.

When we compare these topic results to the results obtained from ukWaC with the same method, the similarity still remains high, but lower than between hrWaC and slWac. ukWac shares 13 similar topics with both, an additional one with slWaC and two with hrWaC.

In other words, (European) web corpora are rather similar to each other, regardless of the language they are produced in, nevertheless showing greater similarity between web corpora of culturally and linguistically more related languages.

4 Conclusion

The paper presented two new Web corpora, for Croatian and Slovene, and the pipeline that was used for building them. The pipeline introduces some changes to the current methods for Web corpus building, especially in the crucial step of content extraction, leading to cleaner corpora. Since the amount of available information in corresponding languages is not as high as for other, larger languages, our method manages to bypass methods known for eliminating a considerable amount of collected data. Further work is necessary to compare the quality of these corpora compared to already existing Web corpora.

Additionally, we analyzed the content of the built corpora via topic modeling and compared them to the ukWaC corpus showing a very high degree of similarity between the Web corpora of linguistically and culturally near languages and an overall high degree of similarity between Web corpora in general.

Further work includes, in the first place, enlarging the Slovene part of the corpus, which now lags behind its Croatian counterpart. The corpora will also be made available via a concordancer, possibly via SketchEngine. The main opportunity, however, for these corpora lies in using them for a series of modeling and extraction tasks, like the one currently underway - building comparable corpora of closely related languages for bilingual lexicon extraction.

References

1. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation 43(3), 209–226 (2009)
2. Guevara, E.: NoWaC: a large web-based corpus for Norwegian. In: NAACL HLT 2010 Sixth Web as Corpus Workshop, pp. 1–7 (2010)
3. Spoustov, D., Spousta, M., Pecina, P.: Building a Web Corpus of Czech. In: Seventh Intl. Conf. on Language Resources and Evaluation, LREC 2010 (2010)
4. Sharoff, S.: Analysing Similarities and Differences between Corpora. In: 7th Conference "Language Technologies", Jožef Stefan Institute, Ljubljana, pp. 5–11 (2010)
5. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: WSDM 2010, pp. 441–450 (2010)
6. Stupar, M., Jurić, T., Ljubešić, N.: Language Identification on Web Data for Building Linguistic Corpora. In: Proceedings of the INFuture 2011 Conference (2011) (in press)
7. Agić, Ž., Tadić, M.: Evaluating Morphosyntactic Tagging of Croatian Texts. In: Fifth Intl. Conf. on Language Resources and Evaluation (2006)

8. Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R.: Massive Multilingual Corpus Compilation: Acquis Communautaire and ToTaLe. Archives of Control Sciences 15(3), 253–264 (2005)
9. Erjavec, T., Krek, S.: The JOS morphosyntactically tagged corpus of Slovene. In: Sixth Intl. Conf. on Language Resources and Evaluation (2008)
10. Erjavec, T.: MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In: Seventh Intl. Conf. on Language Resources and Evaluation (2010)
11. McCallum, A.K.: MALLET: A Machine Learning for Language Toolkit (2002), http://mallet.cs.umass.edu