Getting the Ball Rolling: Producing a Foundation Text for a Universal Corpus of the World's Languages

Liling Tan, Guy Emerson, Susanne Fertmann, Alexis Palmer and Michaela Regneri
Universität des Saarlandes
Campus, 66123 Saarbrücken, Germany
s9sufert@stud.uni-saarland.de, emerson@coli.uni-saarland.de,
liling.tan@uni-saarland.de

Abstract

The study of endangered languages is limited by a lack of data. Existing corpora collections are limited in the range of languages covered, in standardisation, or in machine-readability. This makes the situation even worse for the computational linguist, especially one who would like to take a cross-linguistic or typological approach. We first survey existing efforts to compile cross-linguistic resources, then describe our own approach and give an example application - language clustering. To build the foundation text for a Universal Corpus, we crawled and cleaned texts from several web sources that contain data from a large number languages, and converted them into a standardised form consistent with the guidelines set out by Abney and Bird (2010). The resulting corpus is more easily-accessible and machine-readable than any of the underlying data sources, and represents a significant base corpus for researchers to draw on and add to in the future.

1 Introduction

Abney and Bird (2010) posed the grand challenge of building a Universal Corpus, including all of the world's languages, in a consistent structure.

2 Related Work

Currently multilingual corpora efforts have limited coverage in number of languages and the number of language families the corpora represents; for instance, the OPUS corpus covers over 90 languages (Tiedemann, 2012), i.e. 1.27% of the total number of languages in the world; Leipzig Corpora Collection[1] contains corpora in 230 languages, 3.24% of 7105 living languages (Biemann et al. 2007). Even corpora that boast of linguistic diversity lack in language families coverage; e.g. the linguistically diverse NTU-Multilingual Corpus covers only 7 out of 136 language families (Tan and Bond, 2011). The quintessential solution towards a Universal Corpus is to merge all existing corpora and provide an ubiquitous access interface. To compile the foundation text for the Universal Corpus (uniWaC), we crawled and cleaned web data that contains multilingual texts and merge them with existing corpora collections to form the foundation text for the Universal Corpus.

2.1 WaCky Corpora

Baroni and Bernardini (2004) introduced the BootCaT toolkit to bootstrap specialized corpora and terms from the web. Then, Sharoff (2006) built corpora in various languages by issuing queries for random combinations of frequent words to create a balanced corpus not unlike the British National Corpus. Later, Ferraresi (2007) introduced and discussed the notion of compiling Web as Corpus *"properly"* with the inaugural ukWaC using web-crawled data from a list of seed words. Thereafter, the Crbadn corpora was created for a large number of under-resourced languages (Scannell, 2007). Following which the deWaC (German), itWaC (Italian) and frWaC (French) were compiled (Baroni et al. 2008).

Subsequently, Brunello (2009) stressed the notion of building free corpora from the web using documents released under Creative Commons licenses. To build better quality WaCky corpora, Versley and Panchenko (2012) introduced the notion content-sensitive boilerplate detection in cleaning web corpora.

The initial wave of monolingual WaCky corpora (Baroni et al. 2008) inspired more domain specific WaCky corpora, such as the Korean Web

[1] corpora.uni-leipzig.de

Learner's Corpus (Dickinson et al. 2013) and the Feed Corpus (Minocha et al. 2013). Meanwhile, Web-driven corpora compilations also focused on building large multilingual resources, e.g. Leipzig Corpora Collection (`LCC`) (Biemann et al. 2007), the COrpora from the Web (`COW`) project (Schaefer and Bildhauer, 2012).

Different from previous WaCs built under the Web as Corpus kool ynitiative (WaCky), we did not perform seed query web-searching or web crawling to achieve a balanced resource. Instead the foundation text for uniWaC was built using dedicated site crawling and/or site-dependent cleaning from four data sources, viz. (i) *Online Database of Interlinear Text* (ODIN), (ii) *Omniglot* website, (iii) *Universal Declaration of Human Rights* (UDHR) and (iv) *Wikipedia* dumps. Our emphasis for the uniWaC is based on based on ***universality***; covering as many languages as possible is the first priority.

3 Crawling and Cleaning

Although data size matters in general NLP (Banko and Brill, 2001), *universality* is the utmost priority in NLP for the Universal Corpus initiative (aka Human Language Project; Abney and Bird, 2010; 2011). Hence, the uniWaC approach to data collection is based on focused/dedicated crawling unlike previous WaC word/URL list seeding. We manually selected sites that contains texts from a large variety of languages and wrote scripts to crawl and clean specific pages from the different data sources. The crawling and cleaning was done simultaneously.

The resulting files were saved as plaintext file with `UTF-8` encoding. One directory for each data source and one file per language. The parallel data (from ODIN and Omniglot) are saved as tab-delimited files, one line per sentence, one column per language. The monolingual data (from UDHR and Wikipedia) are saved as newline delimited plaintext file, each line denoting a paragraph or sentence and each document separated by an empty line. The files are saved with the following naming convention to allow easy access, the filename contains the data source and its ISO 639-3 code, e.g. `odin-iii.txt` contains the *ODIN* data for the *Nuosu* language.

3.1 ODIN

The ODIN data is easily accessible in XML format from the online database[2], where data for each language is saved in a separate XML file and the Interlinerized Glossed Texts (IGTs) are encoded in the `<igt><example>...</example></igt>` tags. Each XML file is saved under a filename that matches the respective language code. While cleaning the data, filenames that does not adhere to ISO 639-3[3] were excluded in the uniWaC compilation.

a. o lesu mai
 2sg return here
 '*You return here.*'

Figure 1: IGT adhering to Leipzig Glossing Rule.

```
<igt>
  <example>
    <line>a. o lesu mai</line>
    <line>2sg return here</line>
    <line>'You return here.'</line>
  </example>
</igit>
```

Figure 2: An IGT in ODIN's XML format.

The IGTs from a web document usually follow the Leipzig Glossing Rules, where the first line is the source language text, the second line contains the word/morphemic equivalence and the third line is an English gloss. From the ODIN XML format, an IGT as of Figure 1 will be represented as in an XML snippet as in Figure 2. Eventually, we need to clean and extract (i) the source text '*o lesu mai*' without the preceding index '*a.*' and (ii) the target language gloss without the quotation marks '*You return here*'.

```
<igt>
  <example>
    <line>(69) na-Na-tmi-kwalca-t
    (Foley 1991)</line>
    <line>3sgA-1sgO-say-rise-PERF
    </line>
    <line>'She woke me up'
    (by verbal action)</line>
  </example>
</igit>
```

Figure 3: Another IGT in ODIN's XML format.

[2]http://odin.linguistlist.org/download
[3]http://www-01.sil.org/iso639-3

The primary problem in extracting the source text is lack of consistency of the IGTs from the ODIN data. For instance, Figure 2 uses an alphabet bullet convention, i.e. '*a.*', whereas the IGT in Figure 3 uses a bracketed number indexing convention, i.e. '*(69)*'. In addition, the source line in Figure 3 included the citation to the example, which would be considered as noise to corpus.

Lewis and Xia (2010) noted that it is not uncommon for IGT found in documents to deviant from one uniform convention; when compiling the ODIN data, they attempted a regex based approached and reported 59% F-score. Similar to their regex approaches for extracting IGT, we cleaned the source line with regexes and kept the ODIN data with two levels of 'cleanliness':

- *Cleaner*: Removed (i) all heading and trailing text embedded in square or rounded brackets and (ii) heading double character token ending with bracket or fullstop.

  (i) `^(?\s?\w{1,5}\s*[):.]\s*`

  (ii) `[\[\(].{1,}[\]\)]`

- *Cleanest*: Accepts only source lines without any punctuation.

The original version of the ODIN data contains XML files for 1275 languages, while the cleaner version of ODIN contains IGTs for 1042 languages and the cleanest version contains IGTs for 402 languages. The drop from 1275 to 1042 languages was largely because XXXX XML files from the original ODIN data had used language codes that were not in ISO 693-3 and for XXXX other files, the `<igt>...</igt>` tags were missing.

3.2 Omniglot

The Omniglot crawling and cleaning required more tact and after manual inspection of the www.omniglot.com website, only the *'Useful foreign phrases'* and the *'Tower of Babel'* pages were consistent in their HTML markups that allows easy cleaning. We wrote a crawler script that downloads pages with that contains the following in their URLs:

- `www.omniglot.com/language/phrases/*`
- `www.omniglot.com/babel/*`

The *'Useful foreign phrases'* page contains parallel phrases in embedded within the `<th><tr>...</tr><th>` tags. The English phrase and foreign language phrase are encoded separately with `<td>...</td>` tags. Figure 4 shows an instance of an Omniglot *Useful foreign phrase* page. Since the HTML page was designed as WYSIWYG, several pages have non-textual elements within the `<tr>...</tr>` tags for aesthetics. For example, the non-breaking space, `<td>&nbsp;</td>` in Fig. 4; the extracted text was properly converted into plaintext format using the HTMLParser[4] module, during which the non textual elements would have been cleaned.

The same process of crawling URLs and cleaning was performed on the *'Tower of Babel'* pages and the only difference was that the texts were embedded in a `<ol><li>...</li></ol>` tags instead.

```
<th>
  ...
  <tr>
    <td>I don't understand</td>
    <td>No appo cumpresu nutta</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>I don't understand</td>
    <td>Non d'isco</td>
    <td>&nbsp;</td>
  </tr>
  ...
<th>
```

Figure 4: A *Useful Foreign Phrase* page

The primary problem of the Omniglot data is that to the only information about the language of the data on the page is given in the title of the HTML and its URL. Additionally, only the name of the language was given not the ISO 639-3 code. Even though Omniglot pages contain language meta data in XHTML markup attributes, most pages are encoded with the `lang="en"` attribute because the pages are catered for English speaking audience and the boilerplates (e.g. navigation bar, site banner and footer) on the pages are in English. This causes a problem in file storage and access since it is necessary to save the files in the naming convention, e.g. `omniglot-smo.txt` but the language code is not found any where on the page. To resolve the problem of standardizing, we have

[4] http://docs.python.org/2/library/htmlparser.html

written a conversion script using the code to language mappings from `http://www-01.sil.org/iso639-3/iso-639-3.tab`

3.3 Universal Declaration of Human Rights

Although there was a pre-compiled version of the UDHR data from the Natural Language ToolKit (NLTK) corpora distribution[5], the distribution was laden with encoding problems during their conversion from pdf to plaintext format. Originally, we tried to resolve the encoding problems by automatically identifying the encoding and decoding the textfiles using the `libmagic` library[6] and then re-encoding them to `UTF-8` whenever possible.

However the better solution was found when we chanced upon the UDHR site[7] from the `unicode.org` domain. The UDHR textfiles found on the site was free of encoding problem but there was a boilerplate at the start of each file which records it metadata. It was simply clean by skipping the first four lines of each file.

Although the redistribution of previous compiled data is not scientifically attractive but redistribution/recompilation of data to improve the usability and quality of the data is worth mentioning. For instance, the SETimes corpus first compiled by Tyers and Alperen (2010) was cleaned and redistributed by Ljubesic and Agic (2013).

3.4 Wikipedia

Tapping on the crowd-sourced Wikipedia articles for NLP forms a crucial part of developing data-driven NLP tools and application (citation neeeded). To automatically download the Wikipedia dumps, we have scripted the `wget` with ISO 639-2 language code as a variable (`$I`) in a shell script, as such:

```
$ wget http://download.wikimedia.
org/$Iwiki/latest/itwiki-latest-
pages-articles.xml.bz2
```

One major issue with using the Wikipedia dump is the sheer size and extracting text from a glut of Wikipedia markup. To convert compressed Wikipedia dumps to textfiles, we used the WikiExtractor [8] tool.

After converting them into textfiles, we used the following regexes to delete residual Wikipedia markups and magic words[9].

[5]http://nltk.googlecode.com/svn/trunk/nltk_data/index.xml

[6]http://linux.die.net/man/3/libmagic

[7]http://unicode.org/udhr/d/

[8]http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[9]http://en.wikipedia.org/wiki/Help:Magic_words

3.5 Leipzig Corpora Collection

Do we want to mention this?

4 Copyright Issues

Creative Commons Licence, etc.

5 Representation and Universality

Number of languages, size of corpus, etc.

6 Language Clustering

This would be an example application, if we have time...

7 Future Work

Other corpora we could work on...

8 Conclusion

Summarise what we've done!

References

Steven Abney and Steven Bird. 2010. The Human Language Project: Building a universal corpus of the world's languages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 88–97. Association for Computational Linguistics.