Comparable Corpora BootCaT

Adam Kilgarriff, Avinesh PVS, Jan Pomikálek

Lexical Computing Ltd., Brighton, UK

Abstract

The BootCaT method (Baroni and Bernardini, 2004) has proved a fast, effective and versatile approach to corpus building. The method has been applied to small specialist corpora for finding terminology and translations (as originally envisaged by Baroni and Bernardini), and to large, general corpora, for large numbers of languages. To date it has not been applied multilingually. This is our topic. We describe an implemented tool, Comparable Corpora BootCaT, and a pilot evaluation.

1. Introduction

The BootCaT method (Baroni and Bernardini, 2004) has proved a fast, effective and versatile approach to corpus building. Starting from a set of seed words, tuples (typically triples) of the seeds are randomly generated and sent as a query to a search engine. The pages which the search engine puts at the top of its search hits pages are retrieved, and, after a certain amount of filtering, de-duplicating, and cleaning, you have a corpus. For a bigger corpus, all that is required is a large-enough seed set and more queries to the search engine. The method benefits from all the work that the search engines do to identify relevant, non-spam, text-rich pages. The method has been applied to create small specialist corpora for finding terminology and translations (as originally envisaged by Baroni and Bernardini), and also large, general ones.

To date it has not been applied multilingually. In this paper we describe Comparable Corpora BootCat, a program that takes a set of seed terms for a domain in Language 1 (L1), bootcats an L1 domain corpus, finds corresponding seed terms for Language 2 (L2) and bootcats a matching L2 corpus. We describe challenges and procedures, and present a first pass at a ’bilingual word sketch’ and a pilot evaluation.

2. BootCaT

2.1. Implementations

The implementation of BootCaT that we use throughout is WebBootCaT (Baroni et al., 2006) which provides a web interface, with the BootCaT process running on a remote server. This is in contrast to the original toolset, which was for installation on the user’s computer and for running from the Unix command line or Dos prompt.[1]

The WebBootCaT suite uses Onion [2]for deduplication and Justext [3] for filtering (Pomikalek, 2011).

As the original toolset is a set of open-source perl scripts, they are readily open to customisation by anyone wishing to use them, and there are numerous BootCaT variants in use at various places.

2.2. Uses

Uses of the tools for creating small, specialist corpora, including translation and teaching translation, terminology and teaching terminology and domain lexicography, are quite widespread, though used rather than reported on, so the evidence is anecdotal.

The BootCaT method has also been used to create large, general-language corpora for lexicography and general linguistic research (Sharoff, 2006; **?**; Kilgarriff et al., 2010).

2.3. Parameters

There are numerous parameters to select when running a BootCaT procedure. The ones which can be set in the WebBootCat interface (advanced options) are:

- **File Types:** HTML, RSS, MS-Word, pdf, plain text, any.
- **Creative Commons licence only:** (to address possible copyright concerns).
- **Tuple size:** how many items in the search to be sent to the search engine.
- **Max tuples:** The number of queries to be sent to the search engine.
- **Max URLs per query:** For each query result, how many URLs do we attempt to retrieve (from the top of the search hits list)
- **Sites list:** it is possible to restrict the search to sites, or to domains, for example `.it` for pages with URLs ending in .it only.

There are further options determining which retrieved pages are filtered out, which we do not discuss here (and leave at their default settings in the experiments).

There is also the choice of search engine and, of course, the choice of seeds (or, more generally, the methodology for selecting seeds).

[1] Access to WebBootCaT is available to all registered users of the Sketch Engine service, see http://www.sketchengine.co.uk.

[2] http://code.google.com/p/onion/

[3] http://code.google.com/p/justext/

2.4. Corpus Sizes

One question of interest for potential BootCaT users is: how large a corpus do I get? We ran some experiments using seed terms drawn from wikipedia (see discussion below) for three domains, also for two search engines, three 'sizes' and four languages, with results as in Table 1.
The corpora took between 30 seconds and 15 minutes to create, depending on corpus size.

Our third domain was 'pancreatic cancer'. Results for English and German were similar to the above but for French and Czech, our wikipedia-based method for finding equivalent articles could not be applied because there was no corresponding article.

The Urls figure has a maximum of ten times the 'Queries sent' figure, since we take up to ten URLs from each query. In fact it was a lower number in each case, as, for some of the queries, the search engine offered less than ten hits, or there were duplicates among the hits for different queries. The Urls figures for Czech are lowest, as there were often not ten hits for a query, and different queries in the same domain often pointed to the same URL.

The Docs figure has a maximum of the Urls figure, and they would be the same if all URLs sought were found, and provided text which passed through WebBootCaT's de-duplication procedures and filters for pages which do not appear to contain running text of the language in question. Typically around one third of URLs are not found, or the page that is retrieved is rejected.

Web pages are of sizes that vary by orders of magnitude, and a page with 100,000 words in it can turn a small corpus into a large one, so there is no very dependable relation between 'number of documents' and 'size of corpus'. If we divide the number of words in each corpus by the number of documents contributing to it, to give an average document length for each corpus, the figures vary from 1,400 to 26,000. 26,000 was an outlier: in most cases the average document length was between 2,000 and 8,000 words, with there being more long documents in 'Stradivarius' than 'volcanoes'.

One would expect there to be interactions between search engine, language and domain, as different search engines will have prioritised different languages and types of page, and different domains will tend to have different kinds of page. We note some differences, but the experiment is too small-scale to draw inferences. The two search engines provided comparable sizes of corpora.

2.5. Earlier evaluations

In the context of large, general language corpora, the papers mentioned above (Sharoff, 2006; Kilgarriff et al., 2010) both makes efforts to evaluate the resulting corpora. Here, our focus is on small specialised corpora where the one evaluation we are aware of is (**?**). The authors present a group of trainee translators with a translation task (to translate 'Patient Information Leaflets' as found in drug packets, from English into Italian) with a variety of resources including several bootcatted corpora, with seeds emphasising 'domain' or 'genre'; the two bootcat corpus types were found to be the most useful inputs for the task.

3. Going Multilingual

The core method for producing a multilingual BootCat corpus is

- take a set of seed terms for a domain in L1
- bootcat an L1 domain corpus
- take corresponding seed terms for L2
- bootcat an L2 domain corpus.

We call this Comparable Corpora BootCaT as the resulting corpora will be comparable in the sense of the Building and Using Comparable Corpora workshop series:[4] different languages but similar content.

One question that the outline begs is, how do we find corresponding seed terms across languages?

3.1. Finding corresponding Seeds

We would like the L2 seeds to demarcate the same domain as the L1 seeds. The obvious thing to do is to translate them. This might be done by the user, or by automatic lookup in a bilingual dictionary.

The main problem with the first method is that the user may not know the domain, or the language pair, well enough to do the translation. Also it can be time-consuming, and, from the perspective of system development and evaluation, it introduces a large extra unknown into the process: some people will do it better than others, or from a different perspective. We conclude that it is good to offer users the option of translating seeds themselves, or editing automatically-translated seeds, or simply writing the L2 seeds from scratch, but we also need to offer an automatic method.

The main problem with the dictionary-lookup method is the availability and coverage of dictionaries. We would like to cover a large number of language pairs, but each language pair requires its own dictionary (or, ideally, two, one for each direction). We need to cover technical vocabulary, as that is of most use for building domain corpora, so the dictionaries will need to be big, with good coverage of very many domains. Accessing any one such dictionary typically involves extensive negotiation and we ideally want hundreds.

Two resources covering very many language pairs and directions in a convenient online format are Google Translate and Google Dictionary.

[4] http://comparable.limsi.fr/bucc-workshop.html

Language	**Search Engine**	**Queries sent**	Volcanoes			Stradivarius		
			Urls	**Docs**	**Kwds**	**Urls**	**Docs**	**Kwds**
English	Bing	10	84	51	244	70	46	230
		50	318	180	679	230	150	1230
		250	941	515	1580	808	483	5326
	Yahoo	10	67	39	152	60	47	148
		50	281	176	445	267	196	1071
		250	867	527	1232	937	649	3700
French	Bing	10	79	45	150	74	52	264
		50	246	152	461	225	145	1020
		250	755	506	1445	612	379	3815
	Yahoo	10	79	36	118	82	60	720
		50	285	154	695	257	156	1155
		250	994	527	1737	843	510	2317
German	Bing	10	49	39	112	32	19	126
		50	174	139	236	183	136	1071
		250	460	339	1288	407	279	2511
	Yahoo	10	59	44	88	40	21	36
		50	246	161	609	147	82	142
		250	775	449	2135	446	250	792
Czech	Bing	10	38	24	79	26	16	154
		50	78	47	168	44	24	624
		250	239	73	463	194	102	1210
	Yahoo	10	47	27	96	44	31	44
		50	120	69	153	54	29	75
		250	453	158	535	315	182	1108

Table 1: BootCat corpus sizes (in URLs sought, documents contributing text, and thousands of words) for two domains, two search engines, three numbers of queries sent to the search engine, and four languages. ’Urls’ is number of URLs sought. ’Docs’ is the number of web pages that passed through the filters to contribute a document to the corpus. ’Kwds’ is the final corpus size, in thousands of words.

These two resources differ in several ways. For Google Translate, the expected form of input is a text, and the engine aims to disambiguate each term in the input according to context so we get just one translation. This may be convenient for term-translation, if the set of L1 terms is presented as a text (with suitable delimiters between them) as the terms may mutually disambiguate. Google Dictionary often presents multiple translation candidates, like a standard bilingual dictionary.

Second, many more language pairs are offered for Google Translate than Google Dictionary. As at April 2011 Google Translate was available for 1171 directed language pairs and Google Dictionary, for 50.

Third, they operate on different Terms of Use. Google recently stated:

> Due to the substantial economic burden caused by extensive abuse, the number of requests one may make per day will be limited and the API will be shut off completely on December 1, 2011.[5]

We currently have a variant of CCBC that uses Google Translate, but it seems we shall not be able to use it for long.

Our conclusion on translation-via-dictionary-lookup is that it is good where we have access to a good dictionary, but getting access is a problem, language pair by language pair, and leaves us at the mercy of dictionary providers.

A third route does not translate at all, but uses wikipedia, viewed as a comparable corpus, as input.[6] It exists for 265 languages and is freely available, and it is often possible to find corresponding articles in different languages. In some cases they are translations but more frequently they are not. Where we have a corresponding pair we can find keywords and key terms from the L1 wikipedia article and the L2 wikipedia article and use them as seed words for the BootCaT processes. Note that we use wikipedia for seeding the process, but go outside wikipedia to build the corpus: we do not depend on the wikipedia text for the main phase.

The corpora described for Stradivarius, volcanes and pancreatic cancer were created in this way, and they were the corpora usd inthe evaluation. The method for identifying keywords to use as seeds is a variant of word W is N times

[5] http://code.google.com/apis/language/translate/overview.html

[6] The approach has been suggested by Silvia Bernardini, Federico Zanettin and Federico Gaspari, personal communication.

as frequent in corpus X vs reference corpus Y. We calcualted the score based on the following algorithm.

```
Algorithm Pesudo Code:
Consider X - main corpus, Y - reference corpus.
 - For each word $W_i$ in X
    - Make a frequency list
    - Normalise to per-million
    - add100 to each normalised figure
```

$$Score(W_i) = \frac{Value for corpus X}{Value for corpus Y} \quad (1)$$

Once all the scores are calculated the scores are sorted and top 100 from the list are selected as keywords. These keywords are further used as seeds during the WebBootCat.

4. Automatic Term Recognition and Term Translation Spotting

A principle use for domain-specific corpora is term-finding, as a manual, semi-automatic, or fully automatic procedure. The fully automatic approach, ATR, is a topic with a substantial literature: see (Zhang et. al., 2010) for a recent review and an evaluation of alternative approaches.

We expect our corpora to be used for term-finding, most likely in a procedure where an automatic process proposes candidates which are then accepted or rejected by a person. So it is reasonable to evaluate BootCaT according to how good its corpora are as sources for ATR.

Three relevant observations from ATR are:

- Two distinct dimensions for assessing candidate terms are ’unithood’ and ’termhood’. Unithood (only applicable to multi-word candidates) concerns the extent to which the distinct words in a candidate expression should be treated as a single unit. Termhood concerns the extent to which a candidate belongs to the domain, as distinct from the language in general
- Different domains are very different (so a good procedure in one domain may not be good in another)
- Evaluation is very hard. There is little overlap between different resources. Experts differ. Most evaluation efforts only support limited and local conclusions.

An area neighbouring ATR is ‘Term Translation Spotting’, which makes use of comparable corpora, a field inaugurated in (Fung, 1995).To evaluate CCBC corpora, this area is highly salient. It is also relevant for statistical machine translation, as it is closely related to the SMT challenge of finding sentence-pairs that correspond across languages.

While we use ATR to evaluate BootCaT and CCBC, it has not been the focus of our research. Our focus has been the corpus-building itself. At time of writing our ATR machinery is underdeveloped, so ATR results are not yet as good as the corpora may justify.

5. Bilingual word sketches

CCBC corpora can be used for term-finding, as two matching but independent datasets, and it is likely that this will be their most common use. But perhaps we can do more, offering candidate translation pairs. We have observed that L1 and L2 key term lists often contain translation pairs. Could we find them automatically, offering the user a list of likely translations for each L1 term?

We tried this as follows:
The Sketch Engine already has a collocation-discovery method, based on a grammar to find candidates, and statistics to find the most salient candidates (Kilgarriff et. al., 2004). We view terms as a subset of collocations, and use the existing machinery, with a reduced grammar, as a term grammar. (We excluded some grammatical relations that give rise to collocations but are not considered as giving rise to candidate terms, for example, adverb-modifying-verb.)

We then use a bilingual dictionary to translate all the component words appearing in the L1 term list into L2. We lemmatise the corpus and base the analysis on lemmas (eg dictionary headwords) rather than wordforms throughout. We then apply the ’cross-product’ method first proposed by Gregory Grefenstette to find, for each multiword unit, combinations of their translations which are present in the L2 corpus:

```
for each L1 collocation $<$ a,b $>$
 - for each translation of a : $t_a$
  - for each translation of b : $t_b$
   - see if $< t_a, t_b >$ is in the L2 collocation list
```

If it is, we have a candidate tranlsation pair. For any L1 collocation, there may be 0, 1 or multiple L2 collocations (and vice versa).

We would like to produce ‘bilingual word sketches’. Monolingual word sketches are one-page automatic corpus-based summaries of a word’s grammatical and collocational behaviour and have been in use for lexicography since 1998. It is far from clear how the definition should be extended to cover the two-language case, but a first pass at the bilingual word sketch was prepared using the method above (but only with the single translations for each word that Google Translate provided) and is shown in Figure 1.

6. CCBC: Pilot evaluation

We conducted a first evaluation of CCBC by asking bilingual experts, for a small set of corpora and term candidates, “should this term be in a specialised dictionary for the domain”, and, for the two-language case, for each L1 item, is its translation in the L2 list.

We used eight of the corpora described in Table 1: we selected only the ones that used Yahoo, and only the largest, based on 250 search-engine queries, from each set of three sizes. We used two corpora for each language, one on volcanoes and one on Stradivarius. One of the evaluators also assessed the English and German pancreatic-cancer corpora.

For each corpus we identified 30 keywords and 100 top collocations. The keywords, all single words as opposed to multiwords, were the words that had the highest ratio between normalised frequency in the domain corpus and in a large web-crawled reference corpus for the language. In addition keywords had to occur in at least ten different documents.

The 100 top collocations were identified as the items with the highest scores in the domain corpus (with salience as defined on the Sketch Engine website).
This used the technology for collocation-finding, so the collocations were in fact 3-tupes of <word1, word2, grammatical relation> (or in some cases, 4-tuples with the fourth item being a preposition). A consequence was that the same word-pair sometimes occurred twice in the collocation list, once as, eg, <ice, glacial, modifier> and once as <modified, glacial, ice, modified>. There were 15 such duplications in each of the English files, so once we had de-duplicated, there were just 85 items assessed rather than 100.

Whereas the single words were selected purely on the basis of their termhood, the multi-word candidates were selected purely on the basis of their unithood.

The evaluator was presented with the two-part list (single words, and multiwords) and given four possible answers to the question “should this term be in a specialised dictionary for the domain?” - yes, probably, possibly, no. In the event evaluators almost always used ‘yes’ or ‘no’ and the few ‘probably’ values were treated as ‘yes’ and the ‘possibly’ ones as ‘no’. Then, for the multilingual part, the evaluators were asked to judge, whether each of the ‘good’ terms in the L1 list had a translation amongst the ‘good’ terms on the L2 list. There was one evaluator each for Czech and English, German and English and French and English, called E-Cz, E-De and E-Fr in Table 2. All were language professionals, native speakers in one of their languages and of near-native competence in the other. Not all translators completed all parts of the exercise, hence the blank cells in the table.

6.1. Discussion

It is immediately apparent that the system performed well on the single-word terms and poorly on the multiword ones. The majority of the single-word candidates were good in all cases, with only one bad item in thirty in ’volcanoes-en’. (The same bad item was picked out by all three evaluators.) By contrast, the best result for multiword candidates was under one in three.

Likewise for translations: matched corpora often furnished translation pairs from among the single-word lists, with over half of the lists falling into translation-pairs in a couple of cases. For multiword translations, we do not provide a column in Table 2 for the simple reason that our evaluators, when they looked, did not find any. The column would have contained only zeroes and blanks.

Who	**Wds**	**Trans**	**Mwds**
	Volcanoes, En		
E-Cz	29/30		10/85
E-De	29/30	10/29	16/85
E-Fr	29/30	19/29	24/85
	Stradivarius, En		
E-Cz	19/29		13/85
E-De	26/30	3/26	9/85
	Stradivarius, De		
E-De	16/30	2/16	6/84
	Cancer, En		
E-De	27/30	9/27	
	Cancer, De		
E-De	22/30	10/22	8/90
	Volcanoes, Fr		
E-De	27/30	19/27	5/83

Table 2: Manual evaluation results, by corpus and evaluator. Wds: One-word term candidates assessed as good. Trans: the good single-word terms for which translations were found in the corresponding list for the evaluator’s other language. Mwds: multi-word term candidates assessed as good.

For the English lists that all three evaluators assessed, there was very high agreement on what was good for the single-word items, but low agreement for the multiwords. This is, we believe, because it is a hard judgement for a non-expert in the domain to make (specially outside one’s mother tongue, as E-De said when explaining the blanks in her results).

We believe the reasons for the poor performance on multiwords are, firstly, insufficient care in adopting our collocation grammar to a term grammar, and second, the fact that our multiword canddiate selection was based only on unithood, and not at all on termhood.

This was a small pilot evaluation, and over the coming months we shall be undertaking a more careful evaluation. The pilot has shown us that, as measured by results for single-word terms, our corpora look satisfactory, but we need to adopt lessons from ATR and translation-term evaluation in order to improve performance on multiword candidates.

7. Summary

We have presented CCBC, a suite of methods for ’bootcatting’ comparable corpora. We first reviewed BootCaT, and presented some data on the size of corpora that one might expect to generate with a range of search engines, languages, domains, and query-set sizes. We then discussed various ways for turning BootCaT bilingual. We presented a first draft ’bilingual word sketch’.

We made an initial evaluation of BootCaT and CCBC by considering the corpora that were produced as sources for automatic term recognition, and asking experts to evaluate the candidate term lists. This gave some evidence that the corpora were useful, and many pointers for what we need to do next.

Some of the future work could be to try various other automated term recognition methods which use both unithood and termhood for extraction (Zhang et. al., 2010), to evaluate our method.

We could also automate the process of evaluating the term/multiword translations across various langugages by using the existing translation systems/dictionaries.

8. References

Baroni M. and Bernardini S. 2004. BootCaT: Bootstrapping corpora and terms from the web. *Proceedings of LREC 2004*, Lisbon: ELDA. 1313-1316.

Baroni M., Kilgarriff A., Pomikalek J. and Rychly P. 2006. WebBootCaT: a web tool for instant corpora. *Proceedings of Euralex 2006*, Alessandria: Edizioni dell’Orso. 123-132.

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

S. Bernardini, A. Ferraresi and E. Zanchetta. IN Buccbook. Sharoff ed. (deails to follow)

Fung P. 1995. Compiling bilingual lexical entries from a non-parallel English-Chinese corpus. Proc workshop on very large corpora.

Kilgarriff A., S. Reddy, J. Pomikalek and Avinesh PVS 2010 A Corpus Factory for Many Languages *Proceedings of LREC’10*, Valletta, Malta, May 19-21.

Kilgarriff A., P. Rychly, P. Smrz, D. Tugwell 2004. The Sketch Engine. Proc EURALEX 2004, Lorient, France.

Pomikálek J. 2011. Removing Boilerplate and Duplicate Content from Web Corpora. PhD Thesis, Masaryk University, Brno, Czech Republic.

Sharoff S. 2006. Creating general-purpose corpora using automated search engine queries. In Marco Baroni and Silvia Bernardini, editors, *WaCky! Working papers on the Web as Corpus*, Gedit, Bologna.

Zhang Z., J. Iria, C. Brewster, F. Ciravegna 2010. A comparative evaluation of term recognition algorithms.