Finding Terms in Corpora for Many Languages with the Sketch Engine

Adam Kilgarriff
Lexical Computing Ltd., United Kingdom
adam.kilgarriff@sketchengine.co.uk

Miloš Jakubíček and **Vojtěch Kovář** and **Pavel Rychlý** and **Vít Suchomel**
Masaryk University, Czech Republic
Lexical Computing Ltd., United Kingdom
{xjakub, xkovar3, pary, xsuchom2}@fi.muni.cz

1 Overview

Term candidates for a domain, in a language, can be found by

- taking a corpus for the domain, and a reference corpus for the language
- identifying the grammatical shape of a term in the language
- tokenising, lemmatising and POS-tagging both corpora
- identifying (and counting) the items in each corpus which match the grammatical shape
- for each item in the domain corpus, comparing its frequency with its frequency in the refence corpus.

Then, the items with the highest frequency in the domain corpus in comparison to the reference corpus will be the top term candidates.

None of the steps above are unusual or innovative for NLP (see, e. g., (Aker et al., 2013), (Gojun et al., 2012)). However it is far from trivial to implement them all, for numerous languages, in an environment that makes it easy for non-programmers to find the terms in a domain. This is what we have done in the Sketch Engine (Kilgarriff et al., 2004), and will demonstrate. In this abstract we describe how we addressed each of the stages above.

2 The reference corpus

Lexical Computing Ltd. (LCL) has been building reference corpora for over a decade. Corpora are available for, currently, sixty languages. They were collected by LCL from the web. For the world's major languages (and some others), these are in the billions of words, gathered using SpiderLing (Suchomel and Pomikálek, 2012) and forming the TenTen corpus family (Jakubíček et al., 2013).

3 The domain corpus

There are two situations: either the user already has a corpus for the domain they are interested in, or they do not. In the first case, there is a web interface for uploading and indexing the corpus in the Sketch Engine. In the second, we offer WebBootCaT (Baroni et al., 2006), a procedure for sending queries of 'seed terms' to a commercial search engine; gathering the pages that the search engine identifies; and cleaning, deduplicating and indexing them as a corpus (Baroni and Bernardini, 2004). (The question "how well does it work?" is not easy to answer, but anecdotal evidence over ten years suggests: remarkably well.)

4 Grammatical shape

We make the simplifying assumption that terms are noun phrases (in their canonical form, without leading articles: the term is *base station*, not *the base stations*.) Then the task is to write a noun phrase grammar for the language.

5 Tokenising, lemmatising, POS-tagging

For each language, we need processing tools. While many in the NLP world make the case for language-independent tools, and claim that their tools are usable for any, or at least many, languages, we are firm believers in the maxim "never trust NLP tools from people who don't speak the language". While we use language-independent components in some cases (in particular TreeTagger,[1] RFTagger[2] and FreeLing[3]), we collaborate with NLP experts in the language to ascertain what the best available tools are, sometimes to assist

[1] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[2] http://www.cis.uni-muenchen.de/~schmid/tools/RFTagger/

[3] http://nlp.lsi.upc.edu/freeling/

in obtaining and customising them, and to verify that they are producing good quality output. In most cases these collaborators are also the people who have written the sketch grammar and the term grammar for the language.[4]

6 Identifying and counting candidates

Within the Sketch Engine we already have machinery for shallow parsing, based on a 'Sketch Grammar' of regular expressions over part-of-speech tags, written in CQL (Corpus Query Language, an extended version of the formalism developed in Stuttgart in the 1990s (Schulze and Christ, 1996)). Our implementation is mature, stable and fast, processing million-word corpora in seconds and billion-word corpora in a few hours.

The machinery has most often been used to find <grammatical-relation, word1, word2> triples for lexicography and related research. It was straightforward to modify it to find, and count, the items having the appropriate shape for a term.

7 Comparing frequencies

The challenge of identifying the best candidate terms for the domain, given their frequency in the domain corpus and the reference corpus, is a variant on the challenge of finding the keywords in a corpus. As argued in (Kilgarriff, 2009), a good method is simply to take the ratio of the normalised frequency of the term in the domain corpus to its normalised frequency in a reference corpus. Before taking the ratio, we add a constant, the 'simple maths parameter', firstly, to address the case where the candidate is absent in the reference corpus (and we cannot divide by zero), and secondly, because there is no one right answer: depending on the user needs and on the nature of the corpora, the constant can be raised to give a list with more higher-frequency candidates, or lowered to give more emphasis to lower-frequency items.

Candidate terms are then presented to the user in a sorted list, with the best candidates – those with the highest domain:reference ratio – at the top. Each item in the list is clickable: the user can click to see a concordance for the term, in either the domain or the reference corpus.

[4]Collaborators are typically credited on the 'info' page for a reference corpus on the Sketch Engine website. The collaborations are also often agreeable and fruitful in research terms, resulting in many joint publications.

Term	Frequency	Freq/mill	Score
移動局	1374	2512.5	2442.6
基地局	2324	4249.6	2048.5
無線基地局	1025	1874.3	1787.7
移動端末	702	1283.7	1284.7
無線端末	477	872.2	865.4
無線リソース	430	786.3	780.3
通信端末	435	795.4	716.2
制御部	379	693.0	656.0
送信部	337	616.2	602.8
送信電力	326	596.1	574.7
無線通信	439	802.7	569.2
無線通信端末	304	555.9	556.9
識別情報	309	565.0	539.6
制御情報	298	544.9	528.0
ハンドオーバ	270	493.7	492.7

Figure 2: Term finding results for Japanese, WIPO format.

8 Current status

Languages currently covered by the terminology finding system are sumarized in Table 1.

Language	POS tagger	Ref. corpus
Chinese simp.	Stanford NLP	zhTenTen11
Chinese trad.	Stanford NLP	zhTenTen11
English	TreeTagger	enTenTen08
French	TreeTagger	frTenTen12
German	RFTagger	deTenTen10
Japanese	MeCab+Comainu	jpTenTen11
Korean	HanNanum	koTenTen12
Portuguese	Freeling	ptTenTen11
Russian	RFTagger	ruTenTen11
Spanish	Freeling	esTenTen11

Table 1: Terminology support for languages in Sketch Engine in January 2014. POS tagger is mentioned as an important part of the corpus processing chain. The last column shows the corresponding default reference corpus.

The display of term finding results is shown in Figure 1 for English, for a bootcatted climate-change corpus. Figure 2 shows a result set for Japanese in the mobile telecommunications domain, prepared for the first users of the systemm, the World Intellectual Property Organisation (WIPO), using their patents data, with their preferred display format.

The user can modify various extraction related options: Keyword reference corpus, term reference corpus, simple maths parameter, word length and other word properties, number of top results to display. The form is shown in Figure 3.

9 Current challenges

9.1 Canonical form: lemmas and word forms

In English one (almost) always wants to present each word in the term candidate in its canonical,

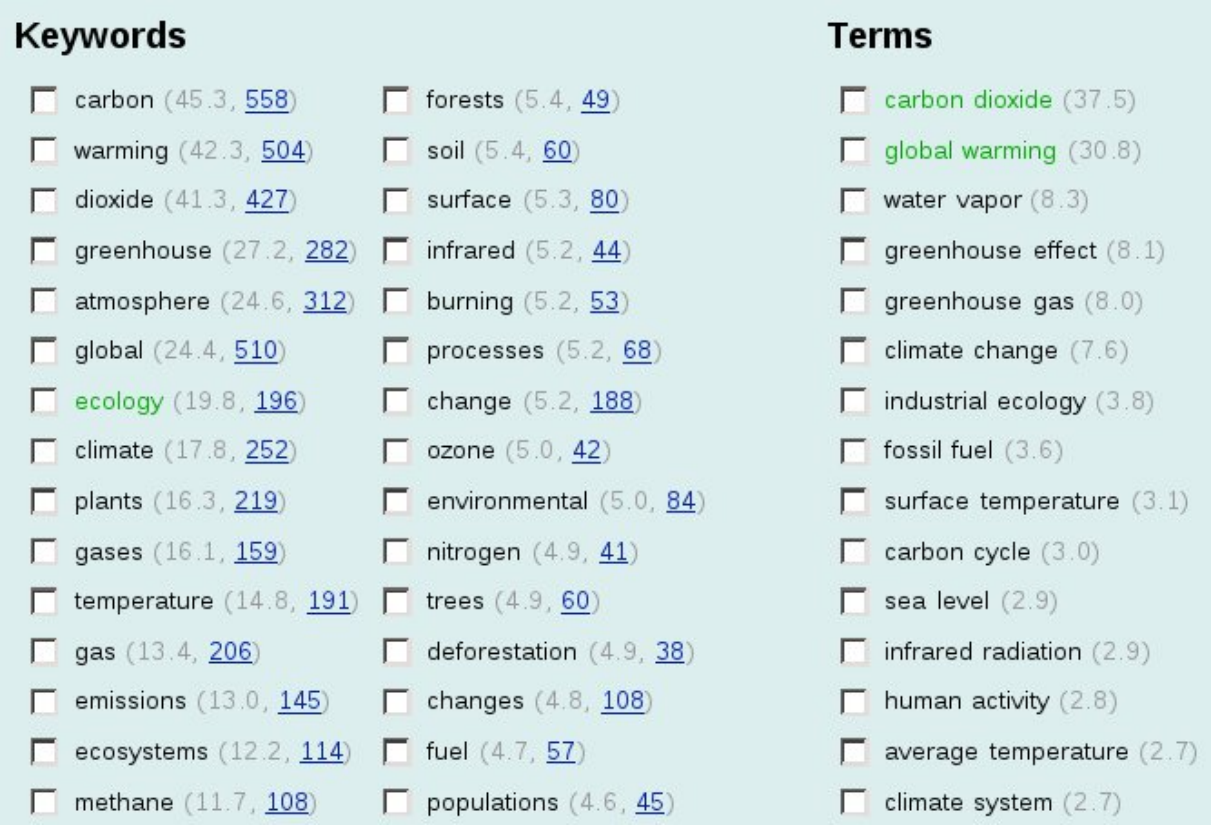

Figure 1: Term finding result in the Sketch Engine – keywords on the left, multiword terms on the right. The values in parentheses represent keyness score and frequency in the focus corpus. The green coloured candidates were used in a WebBootCaT run to build the corpus. The tickboxes are for specifying seed terms for iterating the corpus-building process.

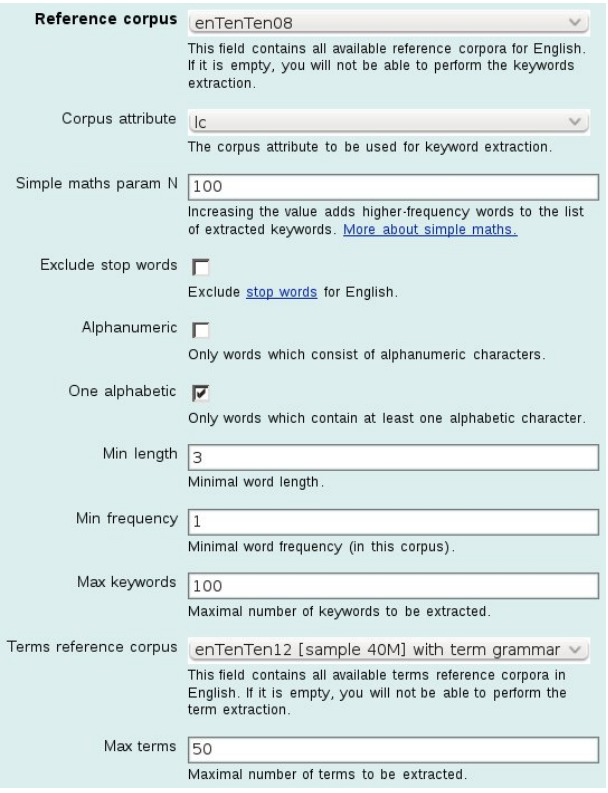

Figure 3: Term finding settings form

dictionary form. But in French one does not. The top term candidate in one of our first experiments, using a French volcanoes corpus, was *nuée ardente*. The problem here is that *ardente* is the feminine form of the adjective, as required by the fact that *nuée* is a feminine noun. Simply taking the canonical form of each word (masculine singular, for adjectives) would flout the rule of adjective-noun gender agreement. A gender respecting lemma turns out necessary in such cases.

Noun lemmas beginning with a capital letter and gender respecting ending of adjectives had to be dealt with to correctly extract German phrases.

In most of the languages we have been working on, there are also some terms which should be given in the plural: an English example is *current affairs*. This is a familiar lexicographic puzzle: for some words, there are distinct meanings limited to some part or parts of the paradigm, and this needs noting. We are currently exploring options for this.

9.2 Versions of processing chains

If the version of the tools used for the reference corpus is not identical to the version used on the

domain corpus, it is likely that the candidate list will be dominated by cases where the two versions treated the expression differently. Thus the two analyses of the expression will not match and (in simple cases), one of the analyses will have frequency zero in each corpus, giving one very high and one very low ratio. This makes the tool unusable if processing chains are not the same.

The reference corpus is processed in batch mode, and we hope not to upgrade it more than once a year. The domain corpus is processed at runtime. Until the development of the term-finding function, it did not greatly matter if different versions were used. For term-finding, we have had to look carefully at the tools, separating each out into an independent module, so that we can be sure of applying the same versions throughout. It has been a large task. (It also means that solutions based on POS-tagging by web services, where we do not control the web service, are not viable, since then, an unexpected upgrade to the web service will break our system.)

10 Evaluation

We have undertaken a first evaluation using the GENIA corpus (Kim et al., 2003), in which all terms have been manually identified.[5]

First, a plain-text version of GENIA was extracted and loaded into the system. Keyword and term extraction was performed to obtain the top 2000 keywords and top 1000 multi-word terms. Terms manually annotated in GENIA as well as terms extracted by our tool were normalized before comparison (lower case, spaces and hyphens removed) and then GENIA terms were looked up in the extraction results. 61 of the top 100 GENIA terms were found by the system. The terms not found were not English words: most were acronyms, e.g. EGR1, STAT-6.

Concerning the domain corpus size: Although the extraction method works well even with very small corpora (e.g. the sample environmental corpus in 1 consists of 100,000 words), larger corpora should be employed to cover more terms. An early version of this extraction tool was used to help lexicographers compile environment protection related terminology. A 50 million words corpus was sufficient in that case. (Avinesh et al., 2012) report 30 million words is enough.

[5]GENIA has also been used for evaluating term-finding systems by (Zhang et al., 2008).

11 Conclusion

We have built a system for finding terms in a domain corpus. It is currently set up for nine languages. In 2014 we shall extend the coverage of languages and improve the system according to further feedback from users.

Acknowledgement

This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

[Aker et al.2013] A. Aker, M. Paramita, and R. Gaizauskas. 2013. Extracting bilingual terminologies from comparable corpora. In *Proc. ACL*, pages 402–411.

[Avinesh et al.2012] PVS Avinesh, D. McCarthy, D. Glennon, and J. Pomikálek. 2012. Domain specific corpora from the web. In *Proc. EURALEX*.

[Baroni and Bernardini2004] M. Baroni and S. Bernardini. 2004. Bootcat: Bootstrapping corpora and terms from the web. In *Proc. LREC*.

[Baroni et al.2006] M. Baroni, A. Kilgarriff, J. Pomikálek, and P. Rychlý. 2006. Webbootcat: instant domain-specific corpora to support human translators. In *Proc. EAMT*, pages 247–252.

[Gojun et al.2012] A. Gojun, U. Heid, B. Weissbach, C. Loth, and I. Mingers. 2012. Adapting and evaluating a generic term extraction tool. In *Proc. LREC*, pages 651–656.

[Jakubíček et al.2013] M. Jakubíček, A. Kilgarriff, V. Kovář, P. Rychlý, and V. Suchomel. 2013. The tenten corpus family. In *Proc. Corpus Linguistics*.

[Kilgarriff et al.2004] A. Kilgarriff, P. Rychlý, P. Smrž, and D. Tugwell. 2004. The sketch engine. *Proc. EURALEX*, pages 105–116.

[Kilgarriff2009] A. Kilgarriff. 2009. Simple maths for keywords. In *Proc. Corpus Linguistics*.

[Kim et al.2003] J-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpusa semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

[Schulze and Christ1996] B. M. Schulze and O. Christ. 1996. The CQP user's manual. *Univ. Stuttgart*.

[Suchomel and Pomikálek2012] V. Suchomel and J. Pomikálek. 2012. Efficient web crawling for large text corpora. In *Proc. WAC7*, pages 39–43.

[Zhang et al.2008] Z. Zhang, J. Iria, C. A. Brewster, and F. Ciravegna. 2008. A comparative evaluation of term recognition algorithms. In *Proc. LREC*, pages 2108–2113.