Corpora by Web Services

Adam Kilgarriff
Lexical Computing Ltd
Brighton, UK
E-mail: adam@lexmasterclass.com

Abstract

Corpora are large objects and querying them efficiently is non-trivial. There are substantial costs to building them, storing them, maintaining them, and building and maintaining software to access them. We propose a model where this work is done by a corpus specialist and NLP systems then use corpora via web services. Our corpus tool is fast, even for billion-word corpora, and offers a wide range of queries via its web API. We have large corpora available for twenty-six languages, and are experts in preparing large corpora from the web, with particular expertise in web text cleaning and de-duplication. We regularly increase our coverage of the world's languages via our 'corpus factory' programme. For English, we are building corpora that are both bigger and more richly marked up than others available. We present a case study of a current project using the Sketch Engine, via its web API, to automatically draft 'fill-the-gap' test items for language testing. The combination of the web services model, the corpora, and the tools, will allow many NLP researchers to use bigger and better corpora in more sophisticated ways than would otherwise be possible.

1. Barriers to entry

In the days of rule-based NLP, starting a PhD was easy. The student could write a few grammar rules, lexical entries and example sentences, and all the technology required was a prolog system.

Since the advent of empirical methods, it is harder. Now the student needs a corpus and tools to access it. Before embarking on their research question - perhaps about syntax, or parsing, or anaphora, or discourse structure - they must first review the different resources they might use, or work out if they must build their own, and then cross the technical and administrative hurdles to building it or acquiring it. They must then either write their own code for accessing it or install and become expert on somebody else's tool. Any output for the first few months is likely to be dominated by aspects of the data or tool that they had not anticipated rather than linguistic ones, and it is all too likely that they start feeling their thesis is being sidetracked into corpora and corpus tools. If they do not have the programming skills or technical support to clear these hurdles, they are likely to become dispirited or to shy away from the question that first motivated them and to switch to one which makes use of corpora in simpler ways, though they may then forever be dogged by the anxiety that their research will not stand up to scrutiny by the researcher, otherwise like them, but who did have the support or computational skill to `do everything properly'.

Might it be possible to use corpora without all this overhead, like a driver collecting a hire car?

We believe not only that it is possible (and that we already have a service offering what is required), but that it is likely to improve the quality of research as energies are not wasted on non-specialist, mediocre, corpus-preparation and corpus-accessing, but are directed at the topic that motivated the researcher.

2. The Sketch Engine

The Sketch Engine is a corpus query tool. It has been widely used for lexicography, by clients including Oxford University Press, Cambridge University Press, Collins, Macmillan and FrameNet, and for linguistic and language technology teaching and research at universities. Corpora for many languages have been installed. It is fast, responding promptly for most queries for billion-word corpora. It offers all standard corpus query functions: concordancing, sorting and sampling of concordances, wordlists and collocates according to a range of parameters, full regular-expression searching, subcorpus specification and searching on subcorpora. It also offers some non-standard ones:

- word sketches: one-page summaries of a word's grammatical and collocational behaviour, see Figure 1
- a distributional thesaurus
- keyword lists which identify the distinctive words of a subcorpus: see Figure 2.

The basic input is a corpus, preferably lemmatised and part-of-speech tagged. For the word sketches and thesaurus, either the corpus must already be parsed, or another input is required: a shallow grammar, written as a regular expression over words and POS-tags, in which each grammatical relation to appear in the word sketch is defined. For a computational linguist with a knowledge of the language in question, preparing a basic grammar is not a large task.

2.1 The Sketch Engine Web Service and API

Lexical Computing Ltd., the owner of the Sketch Engine, provides a web service which gives easy access to corpora. Users can start using the corpus for their question directly: the user interface is simple and there is no software to install.

For four years now there has been a Web API for the Sketch Engine. It is written in JSON and is designed for easy integration into tools written in Java, Python, Perl etc. It covers the core functionality of the Sketch

web **BiWeC freq = 787440**

object_of	33396		and/or	18695		pp_of-i	10574		modifies	649632	
surf	1199	9.79	clipart	503	9.48	deceit	198	7.86	site	264048	11.04
browse	851	8.33	software	2083	6.16	intrigue	176	7.54	page	97315	10.12
weave	629	8.21	correu	36	5.95	spider	95	5.79	browser	19649	9.36
host	1487	7.86	spider	68	5.11	lie	247	5.78	server	14586	8.73
spin	523	7.8	print	372	5.05	interconnection	31	5.66	design	14395	7.96
base	4884	7.71	desktop	115	4.98	deception	67	5.56	cam	4617	7.77
search	1386	7.62	email	509	4.97	interrelationship	21	5.38	designer	5430	7.68
crawl	166	6.66	transience	19	4.97	quill	20	5.34	standard	7949	7.3
scour	116	6.47	designer	213	4.87	interdependence	23	5.26	developer	4218	7.29
chat	256	6.3	gopher	19	4.64	datum	2007	5.0	application	9633	7.11
untangle	83	6.25	telnet	19	4.62	corruption	71	3.99	address	5487	6.99
interconnect	75	5.91	multimedia	63	4.62	trust	126	3.98	interface	3077	6.64

Figure 1: Word sketch for the English noun *web*, drawn from the 5.5b BiWeC corpus, based on 787,440 occurrences (truncated to fit.) The first figure for each collocation is the frequency count, the second is the salience score (Logdice, see help pages at http://www.sketchengine.co.uk). One can sort by either. Other options include 'more data', 'less data' and clustering of collocates. Clicking on the frequency count gives a concordance of the instances.

	new_model_corpus:speech			*new_model_corpus*			
lemma	**Freq**	**ARF**	**ARF/mill**	**Freq**	**ARF**	**ARF/mill**	**Score**
sir	6559	537.5	560.0	8641	1365.5	24.5	16.5
Yeah	12839	1015.8	1058.3	17703	3342.1	60.0	15.3
hey	9294	781.6	814.3	12371	2709.5	48.7	14.1
Hello	4623	411.1	428.3	6574	1392.0	25.0	12.5
okay	9512	709.7	739.4	14043	2997.1	53.8	11.7
fuck	7720	456.6	475.7	11540	1750.3	31.4	11.7
hi	3309	291.0	303.1	4449	961.2	17.3	11.5
shit	4607	370.6	386.1	7266	1554.7	27.9	10.4
No	12340	1097.5	1143.4	20342	5795.3	104.1	10.1
Huh	2667	234.5	244.4	3708	903.8	16.2	9.7
uh	3106	219.2	228.4	4439	828.3	14.9	9.6
oh	17684	1498.8	1561.5	31159	9048.8	162.5	9.1
Bye	1145	102.6	106.9	1284	196.9	3.5	8.6
bye	1233	110.8	115.4	1529	320.0	5.7	8.0
bitch	1643	146.3	152.4	2495	629.8	11.3	7.6
sorry	8127	726.5	756.8	15558	5234.7	94.0	7.4
yes	18823	1667.0	1736.8	37498	12833.4	230.4	7.3
darling	1364	119.9	124.9	1971	495.4	8.9	7.1
honey	1588	140.9	146.8	2689	681.6	12.2	7.1
you	336328	30020.7	31276.7	759009	251027.5	4507.6	6.9

Fig. 2. Top keywords of spoken component of New Model Corpus, as computed and presented in the Sketch Engine, with simple-maths parameter of 10. Component parts (*won, don)* of contracted forms removed.

Engine: one can submit queries which return concordances, word sketches, word lists and thesaurus entries.[1]

2.2 Corpora available in the Sketch Engine

We specialise in large general-language corpora (as required for lexicography). We have publicly-accessible corpora of over 5m words for twenty-six languages (including all major world languages), with over 1 billion words for three: see Table 1.[2] We have a 'Corpus Factory' (Kilgarriff et al 2010) programme for adding to the list of languages in our repertoire by preparing 100m word corpora from web sources, using BootCat methods (Baroni and Bernardini 2004).

Arabic (MSA)	174	Persian	6
Chinese (simp and trad)	456	Portuguese	66
Czech	800	Romanian	53
Dutch	128	Russian	188
English	5,508	Slovak	536
French	126	Slovene	738
German	1,627	Spanish	117
Greek	149	Swedish	114
Hindi	31	Telugu	5
Indonesian	102	Thai	108
Irish	34	Vietnamese	174
Italian	1,910	Welsh	63
Japanese	409		
Norwegian	95		

Table 1: Languages, and the largest corpus available for that language in the Sketch Engine (April 2010, figures in millions of words+punctuation)

3 The Merits of Big, High-Quality Corpora

Since Banko and Brill (2004), it is entirely clear that corpus-based NLP methods tend to perform better, the bigger the corpus. This is one reason for wanting a big corpus. Another is simply to have ample data even for rare phenomena. A third is that a very large corpus will have many large subcorpora. If, for example, we wish to look at Business English, or medical English, or informal English, we can build a classifier to distinguish text of this type from others, and then apply the classifier to a very big corpus, which will then give a subcorpus large enough to support research and model-building for the specific variety.

Corpus quality is less discussed than corpus size. It is harder to define and measure. Also, if people become aware of bad data in their corpus, they are more likely to remove it than announce it. Data cleaning is not high-status work, and papers are likely to pass over it lightly at best, either ignoring the failings of the dataset or presenting results after obvious anomalies have been excluded. Thus a paper which describes work with a vast web corpus of 31 million web pages devotes just one paragraph to the corpus development process, and mentions de-duplication and language-filtering but no other cleaning (Ravichandran, Pantel, and Hovy 2005, section 4). Another paper using the same corpus notes, in a footnote, "as a preprocessing step we hand-edit the clusters to remove those containing non-English words, terms related to adult content, and other webpage-specific clusters" (Snow, Jurafsky, and Ng 2006).

Academic papers do not often present results which compare performance on 'better' and 'worse' corpora. Nonetheless, few would dispute the near-tautology that better corpora are likely to give better results. There are many forms that bad data in corpora can take. They include duplicates, navigation bars and other web material, long lists, logfiles, code, texts in the wrong language, and language-like computer-generated spam. (There are other issues about texts in the correct language but which introduce unwanted biases because there are so many of them. Almost all general-language corpora have this problem, at least from some users' perspective.)

We are corpus specialists. We have explored in depth the issues of web data cleaning (Baroni et al 2008), character encoding (Kilgarriff et al 2010) and de-duplication of large datasets (Pomikalek et al 2009). People accessing our corpora will very often be accessing bigger and better corpora than would otherwise be possible.

3.1 The Google/Yahoo/Bing option

A number of researchers have followed the lead of Grefenstette (1999) and gathered data through extensive querying of one of the main search engines (in Grefenstette's case, Altavista, now usually Google, Yahoo or Bing); see for example Keller and Lapata (2003), Nakov and Hearst (2005), Nakov (2008). The search engines access far more data than we do even in our largest corpora: as against our 5.5 billion, Google indexes at least a trillion words of English. Search engines can be used as a corpus query tool, and if size of data is the overriding consideration, we can offer no alternative. However there are numerous disadvantages to using Google, Yahoo or Bing in this way:

- they are not linguistically aware so do not permit e.g., searches for lemmas
- the query syntax is limited (and subject to change without notice)
- they limit the number of queries one can make
- they limit the number of results per query
- results are sorted according to a scheme which bears no relation to a linguist's wish to see a random sample
- results are not replicable.

(For a full critique, see Kilgarriff 2007.) Using the search engines is a solution with many downsides: if a corpus of 5.5 billion words (for English) is big enough (and for very many, though by no means all, kinds of

[1] Full documentation at http://trac.sketchengine.co.uk/wiki/SkE/Methods/index

[2] We collaborate with numerous groups, and some corpora were built by others, in particular Serge Sharoff at the University of Leeds, UK, and Marco Baroni, Silvia Bernardini, Adriano Ferraresi and colleagues at the Universities of Bologna and Trento, Italy.

research it will be) then there are many advantages to using a specialised service for linguists, such as the Sketch Engine, rather than a search engine.

3.2 New English Corpora 1: BiWeC

BiWeC (Big Web Corpus, Pomikalek et al 2009) is a response to the ongoing need for bigger corpora, and to bridging the gap between corpora that are available in corpus query tools and the web as available via search engine indexes. Our target is 20b words, perhaps 1% of the non-duplicate textual data indexed by Google (see Kilgarriff 2007 for more on relative sizes of large corpora and Google indexes). Our work here has focused on, first, efficient crawling, and then, high-accuracy data cleaning and de-duplication. At time of writing, 5.5 b words have been fully cleaned, de-duplicated, lemmatised, POS-tagged, and loaded into the Sketch Engine.

3.3 New English corpora 2: New Model Corpus

The British National Corpus[3] has been very widely used across linguistics and language technology, and has often been held up as a model for how to design a corpus. However it was designed in the 1980s, before the web existed, and the model, as well as the data, is out of date (for the case in full see Kilgarriff et al 2007).

The next question is: what does a contemporary model corpus look like? The New Model Corpus is a response, comprising 100m words gathered entirely from the web but with proportions of different text types not unlike those of the BNC. It is available for research, and we plan to annotate it as a community-wide exercise, with all NLP researchers invited to download the data, process it with their tools, and return their annotations to us. We shall then integrate the annotations to give a multi-annotated corpus which will also be available for research.

4 A Case Study: TEDDCLOG

TEDDCLOG (Taiwan English Data Driven CLOze test Generation) is a system which drafts fill-the-gap exercises (sometimes known as cloze tests) for learners: the learner is given a sentence in which one word (the *key*) has been replaced with a gap, and a choice of four or five words (the key plus three or four *distractors*) to fill the gap. Exercises of this kind are popular with teachers of English and also with language testing organisations. However the tests are usually based on invented sentences, created by human 'test item writers'. There is a now well-chronicled tendency for there to be a mismatch between the language of invented sentences and that found in corpora of naturally-occurring English.

4.1 The Algorithm

TEDDCLOG uses the following algorithm:

1. User inputs the key
2. Look up the key in the thesaurus to find distractors
3. Find collocates for the key in its word sketch
4. Find a collocate which is used with the key but not with any distractors (the ***koc***, key-only collocate)
5. Find a short simple sentence containing key+koc
6. Prepare output: blank out key from sentence, present key and distractors in random order.

Steps 2-5 each use the web API. They are described in detail below.

4.2 The Corpus

We currently use UKWaC (Ferraresi et al 2008, 1.5 billion words) and may switch to BiWeC.

Size is important for two reasons:

1. A corpus has to be very large to provide more than a handful of sentences for most key-collocate pairings. With more to choose from, there is a better chance that there will be one which is short and simple.
2. It is critical that the distractors are not acceptable alternatives to the key, in the context provided by the sentence. If the corpus is big enough, then the absence of any occurrences of the koc with the distractors is evidence that they are not acceptable.

It would be possible to use a far larger corpus than UKWaC, by using the web as indexed by Google or Yahoo directly. This could give stronger evidence of the non-acceptability of distractors with the koc. (Sumita et al (2005) use a method of this kind.) However the use of the web in this way raises other difficulties as discussed above.

4.3 Worked Example

We want to test the use of the verb *react*. The writer enters *react* into the system.

Finding distractors: the Thesaurus Module

The API call to the thesaurus returns words which typically occur in the same context as the search term. Table 2 shows the SkE Thesaurus for *react*. (The table reveals that most of the words with similar distribution to *react* relate to the human-interaction uses of the word, probably because this is the most frequent kind of use of *react.*) The three top-ranking list members, *respond, interact* and *behave*, are noted and retained for use as PDs (potential distractors).

Finding the Key-only Collocate (Koc): Sketch Differences Module

The Sketch Engine also provides a "Sketch Difference" or *sketchdiff* display, showing which collocates are shared (and "how shared they are") and which are not, between two similar words. Figure 4 shows the sketch differences for *react* and *respond*. We see that *react* occurs 232 times with *positively* as a MODIFIER, and *respond*, 1624 times. The user can click on the number to see the 232, or 1624, concordance lines.

[3] http://natcorp.ox.ac.uk

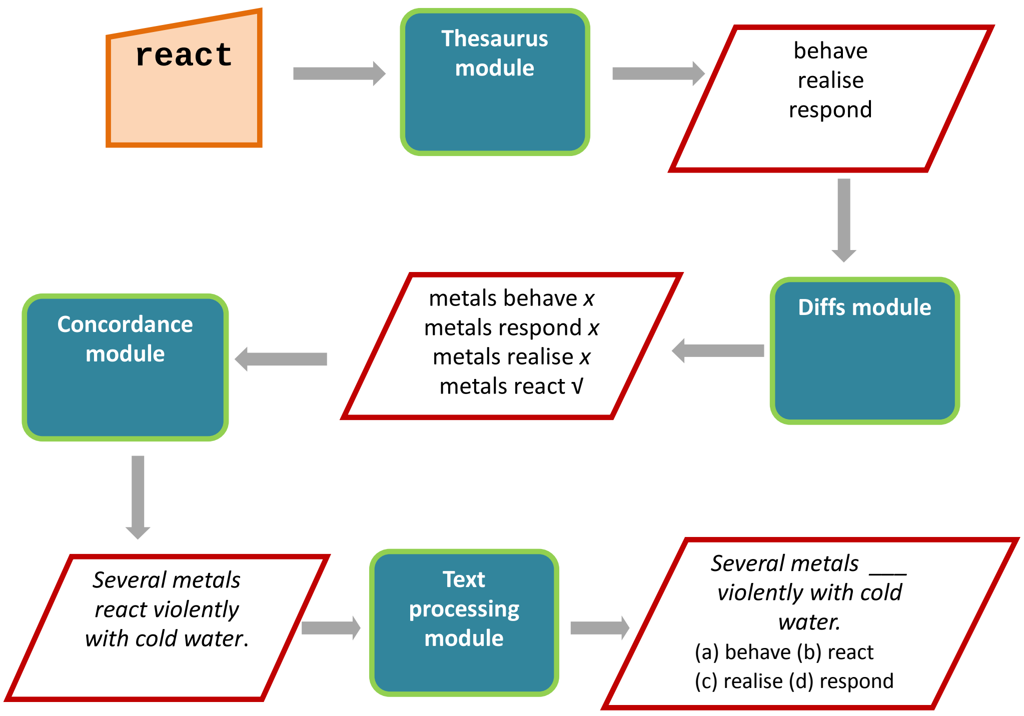
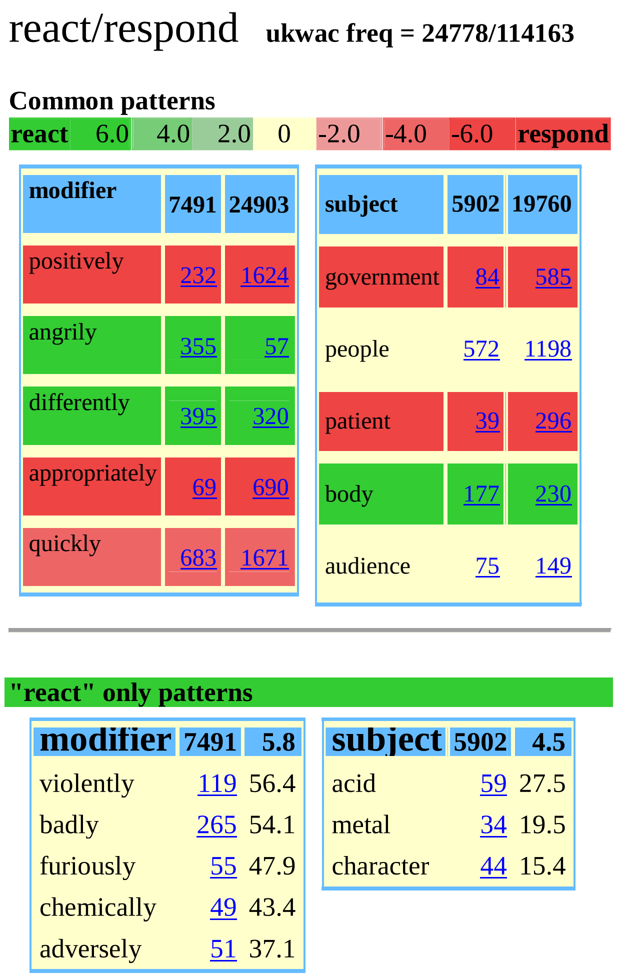


Figure 3. TEDDCLOG System architecture

react **ukWaC freq = 24778**

Lemma	Score	Freq
respond	0.417	114163
interact	0.305	25685
behave	0.296	24508
realise	0.25	110985
cope	0.247	48313
adapt	0.245	50930
listen	0.238	127002
answer	0.237	105714
intervene	0.237	14898
contribute	0.235	137428

Table 2: Distributional thesaurus entry for *react.*



react/respond **ukwac freq = 24778/114163**

Common patterns

react	6.0	4.0	2.0	0	-2.0	-4.0	-6.0	**respond**

modifier	**7491**	**24903**
positively	232	1624
angrily	355	57
differently	395	320
appropriately	69	690
quickly	683	1671

subject	**5902**	**19760**
government	84	585
people	572	1198
patient	39	296
body	177	230
audience	75	149

"react" only patterns

modifier	**7491**	**5.8**
violently	119	56.4
badly	265	54.1
furiously	55	47.9
chemically	49	43.4
adversely	51	37.1

subject	**5902**	**4.5**
acid	59	27.5
metal	34	19.5
character	44	15.4

Figure 4: Sketch Diff for *react* and *respond* (truncated).

TEDDCLOG needs collocates that are not shared with distractors (kocs). Candidate kocs can be seen under the "*react* only" patterns. TEDDCLOG takes the high-salience collocates that do not occur with the first distractor, applying the condition that the collocate must be a correctly spelled English word and not a proper name.

In the simplest case, the first candidate koc does not co-occur with any of the three distractors. In other cases, TEDDCLOG either finds new candidate kocs until one is found that does not occur with any of the distractors, or, depending on parameter settings, finds new distractors from the thesaurus that do not occur with a potential koc. The process is continued until we have a koc, and set of distractors that do not occur with it.

At this point in the algorithm, we have decided on the key and three distractors. We have also established that we wish our carrier sentence to include the collocation: in our example, *metals react.* The next step is to determine what the carrier sentence will be.

Selection of Carrier Sentence

The carrier sentence needs to contain *metal* as subject of *react.* There are 34 such sentences in UKWaC. The next task is to choose the most suitable for a language-teaching, cloze exercise context.

Many sentences are unsuitable, for a range of reasons. For example:

> 2H 2 O 2(aq) == 2H 2 O (l) + O 2(g) or a metal **reacting** with acids, and you can study the effects of a catalyst e.g. adding Cu 2+ (aq) ions to a zinc-acid mixture, though I 'm not sure easy it is to get good quantitative results for advanced level coursework?

Firstly, the sentence is too long, giving the learner work to do which is not directly related to the task that the exercise assesses. Secondly, it contains formulae which will be incomprehensible to non-chemists. Another example is:

> It uses these reactions to explore the trend in reactivity in Group 1. The Facts General All of these metals **react** vigorously or even explosively with cold water .

Here, the problem is that we have not one sentence but two, and a heading and subheading in between. The corpus processing has been led astray by the period following the 1, interpreting it as part of the token "1." rather than as an end-of-sentence marker, and has also failed to mark off the heading ("The facts") and subheading ("General") as not being part of the following sentence.

Atkins and Rundell (2008) discuss the criteria for good examples in dictionary definitions, concluding that such examples must be intelligible to learners, avoiding difficult lexis and structures, puzzling or distracting names, and anaphoric references which cannot be understood without access to the wider context. These lexicographical desiderata are equally applicable to the selection of carrier sentences for cloze exercises. The SkE concordancing software is equipped with a feature called GDEX (Good Dictionary Example Extraction: Kilgarriff et al, 2008), which ranks sentences extracted from corpora according to the following criteria:

- Sentence length: a sentence between 10 and 25 words long is preferred, with longer and shorter ones penalized. (Overshort sentences may not provide enough context to show the user the intended meaning of constituent words.)
- Word frequencies: a sentence is penalized for each word that is not amongst the commonest 17,000 words in the language, with a further penalty applied for rare words.
- Sentences containing pronouns and anaphors like *this that it* or *one* often fail to present a self-contained piece of language which makes sense without further context, so sentences containing these words are penalized.
- Sentences where the target collocation is in the main clause are preferred (using heuristics to guess where the main clause begins and ends, as we do not yet use a parser).
- Whole sentences – identified as beginning with a capital letter and ending with a full stop, exclamation mark, or question mark, are preferred.
- Sentences with 'third collocates', that is, words that occurred with high salience in sentences containing the key and koc, are preferred. This will increase the chances that the context in which the collocation is shown is typical for the collocation.
- Sentences with more than two or three capital letters, and more than two or three punctuation marks and other non-alphanumeric characters, are penalized. This turns out to be a simple way of setting aside most aberrant and junk-filled 'sentences'.

GDEX sorts the concordance lines for any SkE search so that the 'best' sentences are presented first. The sentences which are most likely to be selected for dictionary examples or cloze exercises appear at the beginning of the concordance display. Unwanted sentences, including web noise, are relegated to the end of the concordance so a human user need not waste time looking at them.

TEDDCLOG uses the API with GDEX switched on to find the best sentence containing the key+koc collocation, here *metal* as subject of *react.*

Current status is that we have a prototype system (Smith et al 2009) and are developing a proposal in collaboration with a testing organisation, to turn it into an industrial-strength system.

5 Relation to CLARIN

The EU Project CLARIN[4] aims to establish research infrastructure for language technology, based on web services, and we have approached CLARIN regarding the role that the services discussed here might play. However it seems that CLARIN's perspective is on a longer term and more ambitious plane, with emphasis

[4] http://www.clarin.eu

on standards and community-wide integration, rather than currently-available modest services as here.

6 Summary

We have made the case for 'corpora by web services' with NLP researchers using corpora without needing to store them on their local machines or expend effort on building or maintaining them or associated software. In this way researchers will be able to make use of larger and better corpora than is otherwise possible. The Sketch Engine is a very fast and flexible corpus query tool, into which many large corpora for many languages are already loaded, with a web API, so we are already set for 'Corpora by Web Services' and indeed we already have some users developing NLP applications in this way.

For English, we are developing two new resources with 'Corpora by Web Services' in mind: firstly BiWeC, which moves the scale of resource we offer up by a scale of magnitude, and second, the New Model Corpus, with which we hope to update the BNC as a reference corpus for English. All being well, these two projects will come together in a very large, very well marked up corpus for English which is fully accessible by web API. Using a corpus will not merely be like picking up a hire car, it will be like picking up a Ferrari.

7 References

Banko, Michele, and Eric Brill 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proc ACL*. Toulouse, France.

Baroni, Marco and Silvia Bernardini 2004. BootCaT: Bootstrapping Corpora and Terms from the Web. *Proc LREC*, Gran Canaria.

Baroni, Marco, Francis Chantree, Adam Kilgarriff and Serge Sharoff 2008. CleanEval: a competition for cleaning web pages. *Proc LREC*. Marrakech, Morocco.

Ferraresi, Adriano, Eros Zanchetta, Silvia Bernardini and Marco Baroni 2008. Introducing and evaluating UKWaC, a very large web-derived corpus of English . *Proc. 4th WAC workshop*, LREC, Marrakech, Morocco.

Grefenstette, Gregory. 1999. The WWW as a resource for example-based MT tasks. In *ASLIB Translating and the Computer Conference*, London.

Keller, Frank and Mirella Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.

Kilgarriff, Adam 2007. Googleology is Bad Science. *Computational Linguistics* 33 (1): 147-151.

Kilgarriff, Adam, Sue Atkins and Michael Rundell 2007. BNC Design Model Past its Sell-by. *Proc. Corpus Linguistics*, Birmingham, UK.

Kilgarriff, Adam, Milos Husák, Katy McAdam, Michael Rundell, Pavel Rychlý 2008. GDEX: Automatically finding good dictionary examples in a corpus. *Proc EURALEX*, Barcelona, Spain.

Kilgarriff, Adam, Siva Reddy, Jan Pomikalek 2010. A Corpus Factory for many languages. *Proc LREC*, Malta.

Nakov, P. 2008. Noun compound interpretation using paraphrasing verbs: Feasibility study. *Proc. Artificial Intelligence: Methodology, Systems, Applications* (AIMSA'08).

Nakov, Preslav and Marti Hearst. 2005. Search engine statistics beyond the n-gram: Application to noun compound bracketing. *Proc. Computational Natural Language Learning (CoNLL-2005)*, pages 17–24, Ann Arbor, Michigan.

Pomikalek, Jan, Pavel Rychlý and Adam Kilgarriff 2009. Scaling to Billion-plus Word Corpora. Advances in Computational Linguistics. Special Issue of *Research in Computing Science* Vol 41, Mexico City.

Ravichandran, Deepak, Patrick Pantel, and Eduard Hovy. 2005. Randomized algorithms and NLP: Using locality sensitive hash functions for high speed noun clustering. In *Proc. ACL*, Ann Arbour, Michigan, USA.

Smith, Simon, Adam Kilgarriff, Scott Sommers, Gong Wen-liang, Wu Guang-zhong Automatic Cloze Generation for English Proficiency Testing *Proc. LTTC International Conference on Language Teaching and Testing*, Taipei, Taiwan.

Snow, Rion, Daniel Jurafsky, and Andrew Ng. 2006. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of ACL*, Sydney

Sumita, E., Sugaya, F. and Yamamoto, S. 2005. Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions. *Proc. 2nd Workshop on Building Educational Applications using NLP*, Ann Arbor.