The Sketch Engine

Adam Kilgarriff
Lexicography MasterClass and ITRI, University of Brighton, U.K.

Pavel Rychly, Pavel Smrz
Masaryk University, Brno, Czech Republic

David Tugwell
English Linguistics Department, ELTE, Budapest, Hungary

Abstract

Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour. They were first used in the production of the Macmillan English Dictionary and were presented at Euralex 2002. At that point, they only existed for English. Now, we have developed the Sketch Engine, a corpus tool which takes as input a corpus of any language and a corresponding grammar patterns and which generates word sketches for the words of that language. It also generates a thesaurus and 'sketch differences', which specify similarities and differences between near-synonyms.

We briefly present a case study investigating applicability of the Sketch Engine to free word-order languages. The results show that word sketches could facilitate lexicographic work in Czech as they have for English.

1 Introduction

Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour. They were first used in the production of the Macmillan English Dictionary (Rundell 2002) and were presented at Euralex 2002 (Kilgarriff and Rundell 2002). Following that presentation, the most-asked question was "can I have them for my language?" In response, we have now developed the Sketch Engine, a corpus tool which takes as input a corpus of any language (with appropriate linguistic markup), and which then generates, amongst other things, word sketches for the words of that language.

Those other things include a corpus-based thesaurus and 'sketch differences', which specify, for two semantically related words, what behaviour they share and how they differ. We anticipate that sketch differences will be particularly useful for lexicographers interested in near-synonym differentiation.

In this paper we first provide, by way of background, an account of how corpora have been used in lexicography to date, culminating in a brief description of the word sketches as used in the preparation of the Macmillan dictionary. We then describe the Sketch Engine, including the preprocessing it requires, the approach taken to grammar, the thesaurus, and the sketch differences. We end with a note on our future plans.

1.1 A brief history of corpus lexicography

The first age of corpus lexicography was pre-computer. Dictionary compilers such as Samuel Johnson and James Murray worked from vast sets of index cards, their ‘corpus’.

The second age commenced with the COBUILD project, in the late 1970s (Sinclair 1987). Sinclair and Atkins, its devisers, saw the potential for the computer to do the storing, sorting and searching that was previously the role of readers, filing cabinets and clerks, and at the same time to make it far more objective: human readers would only make a citation for a word if it was rare, or where it was being used in an interesting way, so citations focused on the unusual but gave little evidence of the usual. The computer would be blindly objective, and show norms as well as the exceptions, as required for an objective account of the language. Since COBUILD, lexicographers have been using KWIC (keyword in context) concordances as their primary tool for finding out how a word behaves.

For a lexicographer to look at the concordances for a word is a most satisfactory way to proceed, and any new and ambitious dictionary project will buy, borrow or steal a corpus, and use one of a number of corpus query systems (CQSs) to check the corpus evidence for a word prior to writing the entry. Available systems include WordSmith, MonoConc, the Stuttgart workbench and Manatee.

But corpora get bigger and bigger. As more and more documents are produced electronically, as the web makes so many documents easily available, so it becomes easy to produce ever larger corpora. Most of the first COBUILD dictionary was produced from a corpus of 8 million words. Several of the leading English dictionaries of the 1990s were produced using the British National Corpus (BNC), of 100M words. The Linguistic Data Consortium has recently announced its Gigaword (1000M word corpus) – and the web is perhaps 10,000 times bigger than that, in terms of English language text (Kilgarriff and Grefenstette 2003). This is good. The more data we have, the better placed we are to present a complete and accurate account of a word’s behaviour. But it does present certain problems. Given fifty corpus occurrences of a word, the lexicographer can, simply, read them. If there are five hundred, it is still a possibility but might well take longer than an editorial schedule permits. Where there are five thousand, it is no longer at all viable. Having more data is good – but the data then needs summarizing.

The third age was marshaled in by Ken Church and Patrick Hanks’s inauguration of the subfield of lexical statistics in 1989 (Church and Hanks 1989). They proposed Mutual Information as a measure of the salience of the association between any two words. If, for the word we are interested in, we find all the other words occurring within (say) five words of it, and then calculate the salience of each of those words in relation to the node word, we can summarise the corpus data by presenting a list of its most salience collocates.

The line of enquiry generated a good deal of interest among lexicographers, and the corpus query tools all provide some functionality for identifying salient collocates, along these lines. But the usefulness of the tools was always compromised by:

- the bias of the lists towards overly rare items
- the lists being based on wordforms (*pigs)* rather than lemmas (*pig (noun)).*
- the arbitrariness of deciding how many words to left or right (or both) to consider
- assorted noise, of no linguistic interest, in the list
- the inclusion in the same list of words that might be the subject of a verb, the object of the verb, an adverb, another associated verb or a preposition.

The first issue is one of salience statistics. A number have been put forward, and modern CQSs choose the best, or offer a choice. The second is a matter of, first, lemmatizing the text, and then, applying the lists to lemmas rather than word forms. Here again, various CQSs provide options.

2 The Word Sketch

The word sketch, in addition to using a well-founded salience statistic and lemmatization, addresses the remaining three questions. It does this by using grammar patterns. Rather than looking at an arbitrary window of text around the headword, we look, in turn, for each grammatical relation that the word participates in. In work to date, for English, we have used a repertoire of 27 grammatical relations, for Czech, 23 relations. The word sketch then provides one list of collocates for each grammatical relation the word participates in. For a verb, the subject, the objects, the conjoined verbs (*stand and deliver, hope and pray)*, modifying adverbs, prepositions and prepositional objects, are all presented in different lists. A (truncated) example is presented in Table 1. For each collocate, the lexicographer can click on the collocate to see the corpus contexts in which the node word and its collocate co-occur.

2.1 Corpus query systems

As noted above, Corpus Query Systems play a large role in corpus lexicography. They are the technology through which the lexicographer accesses the corpus. State-of-the-art CQSs allow the lexicographer great flexibility, to search for phrases, collocates, grammatical patterns, to sort concordances according to a wide range of criteria, to identify ‘subcorpora’ for searching in only spoken text, or only fiction. One reading of a word sketch is that it is simply an additional option for accessing the corpus, so should be integrated into a corpus query system to add to the existing armoury of corpus interrogation strategies. This was the how we decided to proceed in developing the sketch engine. We took an existing CQS, Manatee, and added functionality to it.

***pray* (v)** BNC freq= 2455

~ for	**680**	**3.4**	**~ to**	**142**	**1.1**	**and/or**	**179**	**1.7**	**modifier**	**338**	**0.5**	**object**	**183**	**-1.2**	**subject**	**1361**	**0.5**
rain	12	19.8	god	32	24.0	hope	20	20.8	silently	15	13.3	god	13	10.5	we	306	12.3
soul	14	19.3	God	22	17.7	hop	13	15.5	together	35	9.3	God	11	9.6	petitioner	7	8.3
-	117	17.3	lord	16	11.4	fast	6	12.2	fervently	4	7.6	prayer	6	7.6	knee	5	6.9
God	11	16.5	saint	4	10.0	pray	16	11.2	aloud	6	7.5	day	9	3.8	congregation	4	6.8
peace	25	16.5	jesus	2	5.4	kneel	5	9.9	earnestly	5	7.3	heaven	2	3.3	i	263	6.2
miracle	8	13.9	emperor	2	5.2	read	9	9.5	inwardly	3	5.5	hook	2	3.3	she	130	5.8
him	26	13.7	Jesus	2	4.5	talk	6	7.4	hard	7	5.3	time	13	3.2	muslim	3	5.7
forgiveness	7	13.4	spirit	2	4.3	sing	4	6.4	daily	3	4.4	night	5	3.1	follower	3	5.0
you	23	13.2	image	2	4.0	watch	4	5.0	only	20	3.8	lord	2	2.7	Jesus	5	4.8
me	24	13.1	wind	2	3.9	live	3	3.9	continually	3	3.7	pardon	2	2.7	jew	3	4.5
deliverance	6	13.0	him	6	3.3	work	5	3.5	regularly	5	3.5	soul	2	2.4	church	7	4.5
them	23	12.2				wish	2	3.4	often	10	3.3	silence	3	2.4	fellowship	2	4.0
church	12	11.7				believe	2	2.9	ever	9	3.0				Singh	2	3.7
guidance	8	11.6				learn	2	2.8	secretly	2	2.7				Family	6	3.6
us	16	11.6				tell	2	2.3	quietly	3	2.4						
chance	5	10.3							still	11	2.3						

Table 1: Word sketch for *pray (v)*

3 The Sketch Engine

The Sketch Engine is a corpus query system which allows the user to view word sketches, thesaurally similar words, and 'sketch differences', as well as the more familiar CQS functions. The word sketches are fully integrated with the concordancing: by clicking on a collocate of interest in the word sketch, the user is taken to a concordance of the corpus evidence giving rise to that collocate in that grammatical relation. If the user clicks on the word *toast* in the list of high-salience objects in the sketch for the verb *spread*, they will be taken to a concordance of contexts where *toast (n)* occurs as object of *spread (v).*

3.1 Lemmatisation

In order for the word sketch to classify lemmas, it must know, for each text word, what the corresponding lemma is. The Sketch Engine does not support this process; various tools are available for linguists to develop lemmatizers, and they are available for a number of languages (see eg Beesley and Kartunnen 2003). If no lemmatizer is available, it is possible to apply the Sketch Engine to word forms, which, while not optimal, will still be a useful lexicographic tool.

3.2 POS-tagging

Similarly for part of speech (POS) tagging. This is the task of deciding the correct word class for each word in the corpus – of determining whether an occurrence of *toasts* is an occurrence of a plural noun or a 3rd person singular, present tense verb. A tagger presupposes a linguistic analysis of the language which has given rise to a set of the syntactic categories of the language, or tagset. Tagsets and taggers exist for a number of languages, and there are assorted well-tried methods for developing taggers. The Sketch Engine assumes tagged input.

3.3 Input format

The input format is as specified for the Stuttgart Corpus Tools: Each word is on a new line, and for each word, there can be a number of fields specifying further information about the word, separated by lemmas. The fields of interest here are wordform, POS-tag and lemma. The fields are separated by tabs. Constituents such as sentences, paragraphs and documents may also be identified, between angle brackets, on a separate line, as in Table 2 below. (The bracketed word class following the word in the third column for English is one component of the lemma, the other beign the string that forms the word. Thus, for current purposes, *brush (verb)* and *brush (noun)* are two different lemmas.)

```
<s>
The     DET      the (det)
cat     N-sing   cat (noun)
sat     V-past   sit (verb)
on      PREP     on (prep)
the     DET      the (det)
mat     N-sing   mat (noun)
.       PUN      .
</s>
```

```
<s>
Kočka     N-sg-fem-nom        kočka
seděla    V-past-sg-fem-p3    sedět
na        PREP-loc            na
rohožce   N-sg-fem-loc        rohožka
.         PUN
</s>
```

Table 2: Input format

Further information about these constituents can be appended as attributes associated with the constituents. The formalism is fully documented at

http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/.

3.4 Grammatical relations

In order to identify the grammatical relations between words, the sketch engine needs to know how to find words connected by a grammatical relation in the language in question. The sketch engine countenances two possibilities.

In the first, the input corpus has been parsed and the information about which word-instances stand in which grammatical relations with which other word-instances is embedded in the corpus. Currently, dependency-based syntactically annotated corpora are fully supported. Phrase-structured trees need heads of phrases to be marked.

In the second, the input corpus is loaded into the sketch engine unparsed, and the sketch engine supports the process of identifying grammatical relation instances. In this approach, we distinguish two roles: a regular user such as a lexicographer, and an expert user, ideally a linguist with some experience and familiarity with computational formalisms. The expert user will then define each grammatical relation, using the sketch engine to test and develop it, and will load the grammatical relation set into the sketch engine. The sketch engine will then find all the grammatical relation instances and give all users access to word sketches.

The formalism for the grammatical relations is the formalism used for all searches that a user (expert or regular) might make on the corpus. It uses regular expression over POS-tags. An example: if we wish to define the English verb-object relation, we first note that, lexicographically, the noun we wish to capture is the head of the object noun phrase, and that this is generally the last noun of a sequence that may include determiners (DET), numbers (NUM), adjectives (ADJ) and other nouns (N). We also note that the object noun phrase is, by default, directly after the verb in active sentences, and that the lexical verb (V) is generally the last verb of the verb group. Adverbs (ADV) may intervene between verb and object. Taken together, these give a first pass definition for a “verb-object” pair, as “a verb and the last noun in any intervening sequence of adverbs, determiners, numbers, adjectives and nouns”. In the Sketch Engine formalism, using the tags given in brackets above, this is

```
1:″V″ ‶(DET|NUM|ADJ|ADV|N)″* 2:″N″
```

The 1: and 2: mark the words to be extracted as the first and second arguments of the grammatical relation. |, (), and * are standard regular expression metacharacters. | is for disjunction and * indicates that the preceding term (here, the bracketed disjunction) occurs zero or more times.

The expert defines each grammatical relation in this way. Clearly, they need to be conversant with both the tagset and the grammar of the language. As the grammatical relations query language is the standard one for the CQS, they can use the CQS to test grammatical relation definitions and the process of grammatical relation development is well-supported. A definition can have multiple clauses: in our work on English, we have used separate clauses for objects realized as subjects of passives, and nouns which are objects of a verb in a relative clause. Czech sketches define several clauses to capture verbal modifiers in different grammatical cases.

While there are no limits to the sophistication with which one might define a grammatical relation, we have found that very simple definitions, such as the one above, while linguistically unsatisfactory, produce very useful results. While a simple definition will miss grammatically complex instances, it is generally the case that a small number of simple patterns cover a high proportion of instances, so the majority of high salience collocates are readily found, given a large enough corpus. Our use of word sketches to date suggests that POS-tagging errors are more frequently the source of anomalous output than weaknesses in the grammar. The use of sorting based on salience statistics means that occasional mis-analyses rarely result in wrong words appearing in collocate lists.

Verb-object, while frequently the most significant grammatical relation for describing the behaviour of nouns and verbs, is also a relatively complex one to identify. Others such as the relation between an adjective and the noun it modifies (which is usually the most significant one for adjectives) or between a word and others of the same word class that it occurs in conjunction with (*fish/chip; hope/pray; big/fat*), or between a content word and a following preposition, are generally simpler.

These kinds of methods have been widely used; a series of workshops on Finite State methods have been among the places at which Finite State (including regular-expression) approaches to grammatical analysis have been studied. Researchers such as Gahl (1998) have explored sophisticated syntactic querying within a CQS using the same formalism.

3.5 Grammatical relation definitions and free word order

The grammatical relations formalism is sequence-based, and is thereby more obviously suited to languages with a regular word order, such as English, and less clearly suited to a relatively free word order language such as Czech.

For Czech, the defined patterns were based on the grammar employed in SYNT - a robust deep parser for free Czech text (Smrz and Horak, 2000). We started with complex patterns, following the complexity of rules in the grammar, aiming at high precision, and had few mismatches in the retrieved grammatical relations. However, the outcome was a large reduction in the number of identified occurrences of grammatical relations, which resulted in word sketches which were not very informative. So, in a stepwise process, we relaxed constraints, gaining recall at the expense of precision. In this way we found an improved tradeoff between the correctness of the patterns and the usability of the produced sketches.

The current definition of the subject relation for Czech is as below. The keyword DUAL specifies that there are two relations defined here: *is_subj_of* and its converse, *has_subj*, and a single instance of the relation contributes an *is_subj_of* relation to the noun and a *has_subj* relation to the verb. The strings following the equals sign are the names of the relations, separated by a slash. Each line introduces a new clause.

```
*DUAL
=is_subj_of/has_subj
        1:noun_nominative gap([NVZJP].*) 2:[verb_p3X & !aux_verb]
        1:noun_nominative gap([NVZJP].*) 2:[verb_passive & !aux_verb]
        2:[verb_p3X & !aux_verb] gap([NVZJP].*) 1:noun_nominative
        2:[verb_passive & !aux_verb] gap([NVZJP].*) 1:noun_nominative
```

The problem of free word order is addressed by the simple mechanism of gaps in these patterns. The object gap() matches up to 5 words differing in their categories from the given list.

Particular attention has been paid to the agreement constraints that are typical of Czech. Thus, the pattern for adjective modifiers must check that the noun and the corresponding adjective have the same case (c), number (n) and gender (g). The syntax below enforces the match.

```
*DUAL
=a_modifier/modifies
        2:adj adj_string 1:noun & 1.c = 2.c & 1.n = 2.n & 1.g = 2.g
        1:noun 2:adj & 1.c = 2.c & 1.n = 2.n & 1.g = 2.g
```

A 120-million-word Czech corpus, morphosyntactically tagged and automatically disambiguated, has been loaded into the Sketch Engine. Word-sketch patterns have been defined in an iterative process, starting from English-inspired patterns and adding more and more language-specific clauses. We have generated sketches for the 8 875 most frequent Czech words; all those that occurred more than 1 000 times in the corpus.

4 Thesaurus

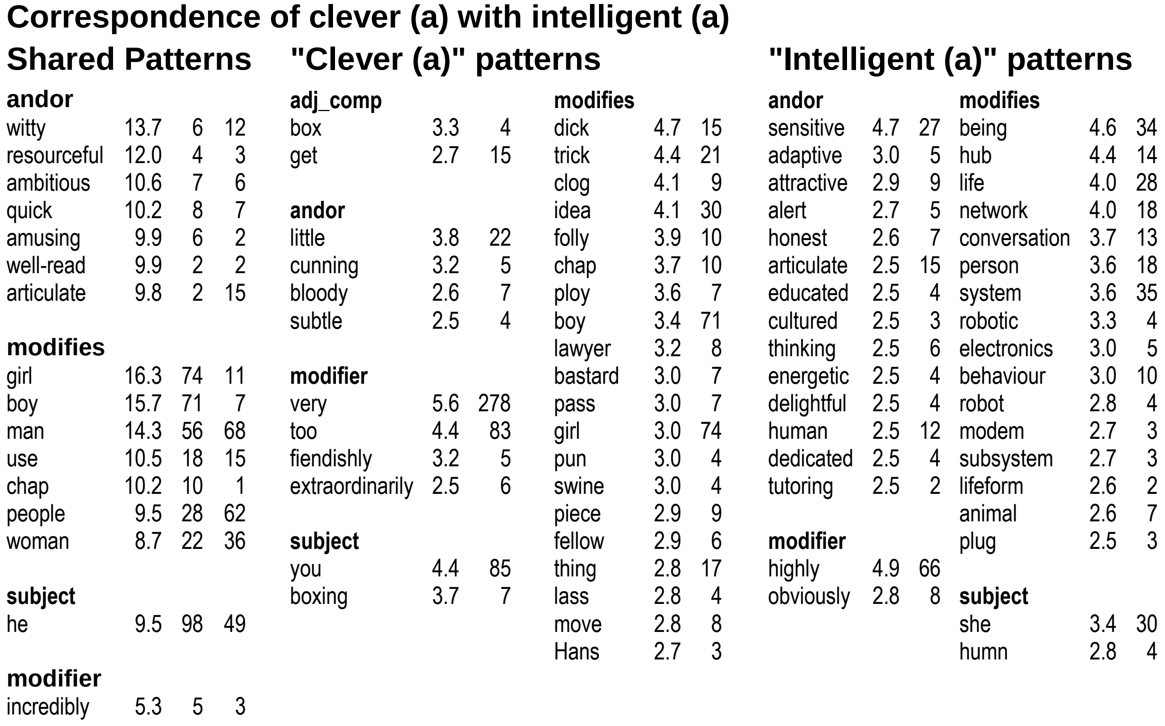
A large set of grammatical relation instances is a rich representation of the lexicon of the language. We can go beyond looking at the behaviour of words one-headword-at-a-time, and use it to show patterns across groups of words.

In particular, when we find the pair of grammatical relation instances <*object, drink, beer*>, <*object, drink, wine*> we can use it as one piece of evidence for *beer* and *wine* being in the same thesaurus category. Here, we are building on a tradition of ‘automatic thesaurus building that goes back to Karen Sparck Jones’s thesis in the 1960s (republished as Sparck Jones 1986) and takes in Grefenstette (1994) and Lin (1998). The Sketch Engine builds a thesaurus, in the form of a set of ‘nearest neighbours’ for each word, using the mathematics for computing similarity as presented by Lin. The thesaurus developed in this way from the BNC is presented in Kilgarriff (2003) and is available to view and to use at http://wasps.itri.bton.ac.uk.

5 Sketch differences

When viewing a thesaurus entry, one repeatedly wonders “what makes those words so similar?”, or indeed, “how do those words differ?” We are in a position to answer this question well. The similarity was based on the ‘shared triples’ (as *beer* and *wine* “share” the triple <*obj, drink, ?*>. What two words have in common are the shared triples that have high salience for both words. The difference between two near-synonyms can be identified as the triples which have high salience for one word, but no occurrences (or low salience) for the other. In the same way that we produced a one page summary as a word sketch, here we can produce a one page summary as a sketch difference.

A pair of near-synonyms we explored using the first prototype sketch difference engine were English adjectives *clever* and *intelligent,* see Table 3. The contrast between the words was immediately apparent. Whereas to call someone intelligent is straightforwardly complimentary, if we call them clever, we may well be implying they are a “clever dick” or “too clever for their own good”. *Clever,* but not *intelligent,* is often found conjoined with *cunning* or preceded by *bloody,* or modifying *swine* or *bastard.*



Correspondence of clever (a) with intelligent (a)

Shared Patterns

andor			
witty	13.7	6	12
resourceful	12.0	4	3
ambitious	10.6	7	6
quick	10.2	8	7
amusing	9.9	6	2
well-read	9.9	2	2
articulate	9.8	2	15
modifies			
girl	16.3	74	11
boy	15.7	71	7
man	14.3	56	68
use	10.5	18	15
chap	10.2	10	1
people	9.5	28	62
woman	8.7	22	36
subject			
he	9.5	98	49
modifier			
incredibly	5.3	5	3

"Clever (a)" patterns

adj_comp		
box	3.3	4
get	2.7	15
andor		
little	3.8	22
cunning	3.2	5
bloody	2.6	7
subtle	2.5	4
modifier		
very	5.6	278
too	4.4	83
fiendishly	3.2	5
extraordinarily	2.5	6
subject		
you	4.4	85
boxing	3.7	7

modifies		
dick	4.7	15
trick	4.4	21
clog	4.1	9
idea	4.1	30
folly	3.9	10
chap	3.7	10
ploy	3.6	7
boy	3.4	71
lawyer	3.2	8
bastard	3.0	7
pass	3.0	7
girl	3.0	74
pun	3.0	4
swine	3.0	4
piece	2.9	9
fellow	2.9	6
thing	2.8	17
lass	2.8	4
move	2.8	8
Hans	2.7	3

"Intelligent (a)" patterns

andor		
sensitive	4.7	27
adaptive	3.0	5
attractive	2.9	9
alert	2.7	5
honest	2.6	7
articulate	2.5	15
educated	2.5	4
cultured	2.5	3
thinking	2.5	6
energetic	2.5	4
delightful	2.5	4
human	2.5	12
dedicated	2.5	4
tutoring	2.5	2
modifier		
highly	4.9	66
obviously	2.8	8

modifies		
being	4.6	34
hub	4.4	14
life	4.0	28
network	4.0	18
conversation	3.7	13
person	3.6	18
system	3.6	35
robotic	3.3	4
electronics	3.0	5
behaviour	3.0	10
robot	2.8	4
modem	2.7	3
subsystem	2.7	3
lifeform	2.6	2
animal	2.6	7
plug	2.5	3
subject		
she	3.4	30
humn	2.8	4

Table 3: Sketch difference for *clever (adj)* and *intelligent (adj)*

We also observe a phenomenon which has been striking in all our thesaurus work: long words tend to go with long words, and short ones with short. *Intelligent* appears to be at home in text types with many long words, whereas *clever* is to be found in less formal genres, amongst shorter words.

We believe the sketch differences provide useful summaries for researchers interested in how pairs of near-synonyms differ.

6 Evaluation of Czech word sketches

The goal for the Czech word sketches was to explore whether it might be possible to substitute standard lexicographic corpus searching by examining only the sketches. We randomly chose 50 words and compared automatically generated sketches with the information given by the biggest two Czech dictionaries (*Slovník spisovné češtiny* and *Spisovném slovníku jazyka ceského*).

Only eight entries contained data that could not be worked directly from the generated sketches. (Idioms in the dictionary were excluded from our comparison here). Moreover, all these cases were generalizations of basic senses that could not be found easily in the corpus and that would probably be missed even with detailed corpus searching. We believe that such results justify future Czech lexicographic projects based on word sketches for the description of the core of the language.

7 Availability, web services

The Sketch Engine is available as a commercial product. It is implemented in C++ and Python. It is designed for use over the web, with a server holding the data and queries issued to the server from a web browser, and with the browser presenting query results. At the time of writing, corpora of Czech, Irish and English have been loaded into the Sketch Engine. The authors are willing to host clients' corpora on their specialist server, and to work with clients on the data preparation.

8 Future plans

1. The software does not yet properly support lexicographic research into multi-word items. When investigating, for example, English phrasal verbs, one would like to explore the grammatical relations and collocations that the phrasal unit entered into. Currently, this is supported only indirectly and minimally. We plan to allow a user to explore a multi-word item (provided it is captured as a grammatical relation triple) as follows. Let us take the English phrasal verb *make up* and assume it is captured as the triple <*following-prep, make, up*>.

The lexicographer first calls up the word sketch for *make* and finds *up* amongst the collocates in the *following-prep* list. They select that preposition and request a word sketch for the triple. The sketch engine then identifies all instances of *make* and *up* occurring in <*following-prep, make, up*>, finds what other grammatical relations those instances participate in, and summarises them in a new 'multiword sketch'.

2. The sketch difference currently contrasts two related words. A comparable task is to look at the same word, in different sets of texts, for example contrasting the use of a word in texts from one era and another, or in written and spoken texts, or in 'original' texts and in translations. We plan to extend sketch difference functionality to make comparisons based on different subcorpora possible.

References

Beesley, K. R. and Karttunen L. *Finite-State Morphology*. Center for the Study of Language and Information (CSLI). 2003.

Church, K. W. and Hanks, P. Word association norms, mutual information and lexicography. Proc. 27th Annual Meeting of ACL, Vancouver. 1989: 76-83.

Gahl, S. Automatic Extraction of subcategorization frames for corpus-based dictionary-building. Proc EURALEX 1998, Liège. 1998: 445-452.

Grefenstette, G. Explorations in Automatic Thesaurus Discovery. Kluwer 1994.

Kilgarriff, A. 2003. Thesauruses for Natural Language Processing. Proc NLP-KE, Beijing, China. Oct 2003

Kilgarriff, A. and Grefenstette, G. Web as Corpus: Introduction to the Special Issue. *Computational Linguistics* 29 (3). 2003.

Kilgarriff A. and Rundell, M. Lexical Profiling Software and its lexicographic applications: a case study. Proc. EURALEX 2002, Copenhagen. 2002: 807-818.

Lin, Dekang. Automatic retrieval; and clustering of similar words. Proc. COLING-ACL Montreal 1998: 768-774.

Rundell, M. Ed. *Macmillan English Dictionary for Advanced Learners*. Macmillan 2002.

Sinclair, J. M. (editor). Looking Up: an account of the COBUILD project in lexical computing. Collins, 1987.

Smrž, P. and Horák, A. Large Scale Parsing of Czech. In Proceedings of Efficiency in Large-Scale Parsing Systems Workshop, COLING'2000. 1st ed. Saarbrucken : Universitat des Saarlandes, 2000: 43-50.

Sparck Jones, K. *Synonymy and Semantic Classification.* Edinburgh University Press. 1986.