Corpus tools for lexicographers

Adam Kilgarriff and Iztok Kosem

1 Introduction

To analyse corpus data, lexicographers need software that allows them to search, manipulate and save data, a 'corpus tool'. A good corpus tool is key to a comprehensive lexicographic analysis – a corpus without a good tool to access it is of little use.

Both corpus compilation and corpus tools have been swept along by general technological advances over the last three decades. Compiling and storing corpora has become far faster and easier, so corpora tend to be far larger. 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. Current lexicographic projects we are involved in are using corpora of around a billion words – though this is still less than one hundredth of one percent of the English language text available on the web (cf. Rundell, this volume).

The amount of data to analyse has thus increased significantly, and corpus tools had to be improved to assist lexicographers in adapting to the change. Corpus tools had to be faster, more multifunctional, and customisable. In the COBUILD project getting concordance output took a long time and then concordances were printed on paper and handed out to lexicographers (Clear, 1987). Today, with Google as a point of comparison, concordancing needs to be instantaneous, with the analysis taking place on the computer screen. Moreover, bigger corpora offer much higher numbers of concordance lines per word (especially for high frequency words), and considering the time constraints of the lexicographers (cf. Rundell, this volume), new features of data summarization are required to ease and speed the analysis.

In this chapter, we review the functionality of corpus tools used by lexicographers. In section 2, we discuss the procedures in corpus preparation that are required for some of these features to work. Then, in section 3, we briefly describe some leading tools, comparing and contrasting them a little. In section 4, we focus first on basic features, which are used also by non-lexicographers, and then move on to the features that are targeted mainly at lexicographers. Section 5 is dedicated to the user-friendliness of corpus tools, a topic that, although rarely discussed in the literature, is becoming more relevant as corpus tools become more complex. Finally, we conclude by considering how corpus tools of the future might be designed to assist lexicographers even further.

2 Preparing the corpus for automatic analysis

Many features of corpus tools work only if the corpus data is properly prepared. The preparation of a corpus has two parts: preparing the metadata, or 'headers', and preparing the text.

A corpus is a collection of documents and instances of a word come from different documents of different types of text. The lexicographer examining the instances of a word may want to know which kind of text a particular instance is from, i.e. from which document, and the characteristics of that document, such as date of publication, author, mode (spoken, written), domain, etc of the text. For this to work, each document must come with metadata, usually located in a 'header', which states features of the document, in a way that the corpus tool can interpret. Using headers, corpus tools can not only provide information on the texts, but also use them to limit the searches to particular text types, build wordlists and find keywords for a text type, and so forth.

Preparing the text starts with identifying and managing the character encoding and then typically involves marking up the text with

1. sections, paragraphs and sentences

2. tokens
3. lemmas
4. part-of-speech tags
5. grammatical structure.

Each text comes with its character encoding. This is the way in which each particular character is encoded in a series of ones and zeros. Widely used character-encodings include ASCII, ISO 8859-1 (also called latin-1), Big-5 (for Chinese) and UTF-8. There are many different character-encodings, most of which are language-specific or writing-system-specific, and they can create a wide range of problems of misinterpretation where one system assumes one encoding has been used, but in fact a different one has. In Latin-script languages, problems most often arise with accented and other non-standard characters since standard characters (a-z, A-Z, 0-9 etc) are encoded in the same way in most encodings. Over time, a growing proportion of documents are encoded using UTF-8, which is based on the Unicode standard, but most documents do not yet use Unicode or UTF8 and the character encoding typically has to be guessed, with each text then converted to the same, standard, encoding.

Sentence, paragraph and section markup (using structural tags) supports functionality such as display of sentences, or not seeking patterns spanning sentence ends. Tokenisation is the process of identifying the tokens, typically the words, which the user typically searches for. For some languages such as Chinese and Arabic this is a major challenge, since for Chinese there is no whitespace between words, and for Arabic many grammatical words are written as clitics, without whitespace between them and the core word. For English it is not a great challenge since, most of the time, whitespace reliably indicates a word break: there are just a few difficult cases, mostly relating to apostrophes (e.g. whether *don't* is counted as one

token or two – *do* and *n't*) and hyphens (*co-operate*, *first-hand*). How a text has been tokenised has an effect on searching, filtering, sorting and many other features.

Lemmatisation (also known as morphological analysis) is (at its simplest) the process of identifying the base form of the word (or the dictionary headword) called a lemma. In a language such as English, many corpus words may be instances of more than one lemma. Thus *tricks* may be the plural of the noun, or the present tense, third person singular form of the verb. The process of identifying, by computer, which part of speech applies in a particular context is called part-of-speech (POS) tagging. Finally, parsing is used to annotate the syntactic structure of each sentence in the corpus.

Once all words in a corpus are lemmatised and part-of-speech tagged (and this information is made available to the corpus tool), each word in the corpus can be thought of as a triple, <word form, lemma, POS-tag>, and searches can be specified in terms of any of these.

In addition to simple searches for single words, lexicographers may often want to search for a phrase or some other more complex structure. A good corpus tool will support complex searches, such as searches by surrounding context, while keeping the interface simple and user-friendly for the simple searches that users most often want to do.

Another form of search uses a corpus query language (CQL), such as the one developed at the University of Stuttgart (Christ, 1995). It allows one to build sophisticated structured searches, matching all- or part-strings, for as many fields of information as are provided (to date, we have seen *word form*, *lemma* and *POS-tag*).

3 An overview of corpus tools

The number of corpus tools has grown over the past thirty years, as not only lexicographers, but also researchers from other linguistics subdisciplines have become aware of the potential of corpora. As these researchers have been interested in many different aspects of language,

corpus tools have become more diverse. Some leading corpus tools have been designed around the needs of a particular institution, project, and/or corpus or corpora, and are tailored for working well in that environment.

Corpus tools can be categorized using the following typology:

a) **Computer-based (standalone) tools vs. online tools**. Some tools work with a model of the corpus and tools being on the user's computer. Leading players here are WordSmith Tools (Scott, 2008) and MonoConc Pro (Barlow, 2002), both of which have been widely and successfully used in teaching. WordSmith and MonoConc Pro are both commercial projects: a free alternative that works in similar ways is Antconc (Anthony, 2011). On the other hand, online corpus tools allow the users to access the corpus, or corpora, from any computer. Examples of online tools include the Sketch Engine (Kilgarriff et al., 2004), KorpusDK (developed by the Department for Digital Dictionaries and Text Corpora at the Society for Danish Language and Literature), and Mark Davies' tools at http://corpus.byu.edu.

b) **Corpus-related tools vs. corpus-independent tools**. Some corpus tools can be used only with a particular corpus, most often because they were designed as a part of a specific corpus project or for a specific institution. Examples include SARA (and its newer XML version, XAIRA) and BNCWeb, two high-specification interfaces designed to access the British National Corpus (BNC), and a tool offered by Real Academia Española to access their Spanish reference corpus, Corpus de Referencia del Español Actual (CREA).[1] A special group of corpus-related tools are tools that use the same interface to access several different preloaded corpora, e.g. the tool KorpusDK that is used to access several Danish corpora. Similarly, corpus tools and software developed by Mark Davies, at Brigham Young University, are used to access leading corpora for Spanish, Portuguese and American English. His websites are

among the most used corpus resources, particularly his Corpus of Contemporary American (COCA) (Davies, 2009). Other tools are corpus-independent, which means that users can use the tools to upload and analyse any corpus they want. These tools include the Sketch Engine, Corpus WorkBench, WordSmith Tools, MonoConc Pro, and AntConc.

c) **Prepared corpus vs. web as corpus**. The majority of corpus tools are used to access a corpus that has been compiled with linguistic research in mind, so is a corpus in a traditional sense of the word. But the web can be viewed as a vast corpus, with very large quantities of texts for many languages, and lexicographers frequently use it in this way (Kilgarriff and Grefenstette, 2003). Google and other web search engines can be viewed as corpus tools: in response to a query, they find and show a number of instances of the query term in use. They are not designed for linguists' use but are often very useful, having access, as they do, to such an enormous corpus. Some tools have been developed which sit between the search engine and the user, reformatting search results as a concordance and offering options likely to be useful to the linguist. They have been called web concordancers. One leading system is Webcorp (Kehoe and Renouf, 2002).

d) **Simple tools vs. advanced tools**, depending on the number of different features provided. Due to the increasing size of corpora, and the increasing number of (different) users, corpus tools have become more and more multifunctional, i.e. they have started offering many different features to assist their users with analysis. The features of corpus tools range from basic features, e.g. concordance, collocation, and keywords, to advanced features, such as Word sketches and CQL search. Most of these features are discussed in more detail in section 4; for more on keywords, see Scott (1997) and Scott & Tribble (2006). Examples of simple corpus tools are

AntConc and MonoConc Easy (Barlow, 2009). Advanced corpus tools are designed for users who need access to more advanced functionality, e.g. lexicographers. Examples of advanced corpus tools are the Sketch Engine, XAIRA, and KorpusDK.

e) **Typical users**. Three main types of users of corpus tools are lexicographers, linguistics researchers and students, and language teachers and learners. Different tools have been designed with different target users in mind.

There are numerous corpus tools, but few with the full range of functionality that a lexicographer wants. Of these, most have been in-house developments for particular dictionary or corpus projects. The tools developed within the COBUILD project were used for lexicography at Collins and Oxford University Press through the 1980s and 1990s and also with the 'Bank of English' corpus and WordBanks Online web service (Clear, 1987). They set a high standard, and have only recently been decommissioned despite using a 1980s pre-Windows, pre-mouse interface.

The University of Stuttgart's Corpus WorkBench, sometimes also called 'the Stuttgart tools', was another influential early player, establishing in the early 1990s a very fast tool suitable for the largest corpora then available, and which could work with sophisticated linguistic markup and queries. It was available free for academic use. Both the format it used for preparing a corpus, and the query language it used for querying a corpus, have become de facto standards in the field. The group that prepared the corpus worked closely with several German dictionary publishers, so the tools were tested and used in commercial lexicographic settings.

As corpora have grown and web speeds and connectivity have become more dependable, computer-based corpus tools have become less desirable for large lexicography projects since the corpus and software maintenance must be managed for each user's

computer, rather than just once, centrally. Consequently, most lexicographic projects nowadays use online corpus tools that use http protocols (so users do not have to install any software on their computer) and work with corpora of billions of words. The Sketch Engine, an online tool developed by the first author's company and used in the second author's projects, has become a leading tool for lexicography and other corpus work since its launch in 2004. The Sketch Engine uses the formalisms and approach of the Stuttgart tools; it is available as a web service, and there are already loaded within it corpora for forty languages. Its other distinctive feature, its use of grammar, is discussed in section 4.2. The tool and its functionality are presented in more detail in the next section.

Recently, lexicographers have become interested in the potential of the world wide web for their data analysis, and consequently also in web concordancers. However, web concordancers rely heavily on search engines which is problematic in various ways, for example there is a limit (for Google, 1000) on the number of hits the user has access to for any search, the corpus lines are sorted according to the search engine's ranking criteria, etc. There are also those who question the lexicographic potential of the web due to its constantly changing size and contents. The debate is ongoing but considering that the web makes so many documents easily available, it would be a shame to not utilize such a resource.

4 Moving on from concordances: the Sketch Engine

The number of features offered by corpus tools is continuously increasing, and a development of a new feature is often the result of a certain lexicographer's need. Recently, many new features have been introduced in the Sketch Engine, a tool aimed particularly at lexicography, and which is available for use with corpora of all languages, types and sizes. The Sketch Engine has had a steady program since inception of adding functionality according to lexicographers' and corpus linguists' needs.

This section focuses on different features of the Sketch Engine, with particular attention being paid to the features used extensively by lexicographers. Many features, especially the ones presented in section 4.1, are found in most corpus tools and should not be considered Sketch Engine-specific. It should also be pointed out that while each new feature is normally used extensively by lexicographers, it later becomes widely used by linguists, educators and other researchers. In view of that, the features presented in this section should not be regarded as lexicographic, even though some of them have (so far) mainly been used in dictionary-making.

4.1 Analysing concordance lines

The **concordance,** "a collection of the occurrences of a word-form, each in its textual environment" (Sinclair, 1991: 32), is the basic feature for using a corpus, and is at the heart of lexicographic analysis. Concordance lines can be shown in the sentence format or in the KWIC (Key Word in Context) format. The KWIC format, preferred in lexicography, shows a line of context for each occurrence of the word, with the word centred, as in Fig. 1. Using the concordance feature, lexicographers can scan the data and quickly get an idea of the patterns of usage of the word, spotting meanings, compounds etc.

<<Fig. 1>>

The problem with reading raw concordance data is that it can be very time-consuming for lexicographer to gather all the required information on the analysed item. Lexicographer may also want to focus on a particular pattern found in the concordance, group similar concordances together, etc. It is therefore useful for the lexicographer to have available additional features that help manipulate the concordance output and give some statistical information on it. Some of these features are presented below.

Sorting the concordance lines will often bring a number of instances of the same pattern together, making it easier for the lexicographer to spot it. The most typical sort is

sorting by the first word to the left or first word to the right, and sorting by the node word. Sorting by the node word can be useful for lexicographers working with highly inflected languages where lemmas often have many different word forms. The type of sorting that yields more useful results depends on the grammatical characteristics of the word; for example, nouns, sorting to the first word on the left will normally highlight the relevant patterns involving adjective modifiers and verbs that the noun is object of, whereas sorting to the right will show verbs that the nouns is subject of. In Fig. 2, where the concordance lines are sorted to the first word to the right, it is much easier to spot recurring patterns such as *argue for* and *argue that*, as opposed to Fig. 1. Other types of sorting include sorting according to the second, third, etc. word to the right or to the left of the node word, and more complex options such as sorting according to word endings.

<<Fig. 2>>

There are two more types of sorting that differ from the types of sorting mentioned so far, namely sorting according to the meaning of the node word, and sorting according to how good of a candidate for a dictionary example the concordance line is. Both types require an additional stage before the sort can be performed – the former requires manual annotation of the concordance lines of the word (see section 5.4), whereas the later requires the computation of the good example score (see section 4.3).

Sampling is useful as there will frequently be too many instances for the lexicographer to inspect them all. When this is the case, it is hazardous just to look at the first ones as they will all come from the first part of the corpus. If the lexicographer is working on the entry for *language*, and there are a few texts about *language development* near the beginning of the corpus, then it is all too likely that the lexicographer gets an exaggerated view of the role of that term, while missing others. The sampling feature in the corpus tool allows the

lexicographer to take a manageable-sized sample of randomly selected concordance lines from the whole corpus.

Filtering allows the lexicographer to focus on a particular pattern of use (a positive filter), or to set aside the patterns that have been accounted in order to focus on the residue (a negative filter). For example, if the lexicographer spots that *local authority* as a recurrent pattern of the word *authority,* he can first focus on that pattern by using either the positive filter (searching for all the concordances where *local* occurs one word to the left of *authority*), or performing the search for the phrase *local authority*, and then continue the analysis by excluding the pattern *local authority* from the concordance output with the negative filter.

Search by subcorpora can be considered as a type of filtering as it can be used to limit the analysis of the pattern to part of the corpus. Many words show different meanings and patterns of use in different varieties of language, and the lexicographer needs to be able to explore this kind of variation. A vivid example is the English noun *bond:* in finance texts it means a kind of finance, as in *treasury bonds, Government bonds, junk bonds;* in chemistry, a connection between atoms and molecules as in *hydrogen bonds, chemical bonds, peptide bonds,* and in psychology, a link between people: *strengthening, developing, forging bonds.*

Frequency analyses are often useful to lexicographers. A case combining analysis by text type and change over time using the Sketch Engine's frequency feature is *random.* The goal here was to explore the hypothesis that it has recently added an informal use to its traditional, formal and scientific one, as in

(1) Last was our drama but unfortunately our original drama went down the drain way down so Iffy came up with one very **random** drama involving me doing nothing but

just sit down and say my one and only line " Wha ? " and she just yell at me coz she was pissed off of something.

The Oxford English Corpus (OEC), containing over 2 billion words, contains a large component of blog material, so the blog subcorpus could be used to explore the new pattern of use. Also each text has the year in which it was written or spoken in its metadata. Fig. 3 shows the frequency distribution of the word *random* in blogs over the period 2001-2005.

<<Fig. 3>>

Sometimes the lexicographer cannot decipher the meaning of the analysed word because the concordance line does not provide enough information. For example, for the concordance line for *random* offered above, the default Sketch Engine context size of 40 characters to the left and to the right of the searched word does not provide enough information to get an idea of the meaning of *random*:

(2) drain way down so Iffy came up with one very random drama involving me doing nothing but just sit

It is thus useful to have quick access to more context, which in most corpus tools can be accessed by clicking on a concordance line.

Moving on from concordances

Since COBUILD, lexicographers have been using KWIC concordances as their primary tool for finding out how a word behaves. But corpora get bigger and bigger. This is good because the more data we have, the better placed we are to present a complete and accurate account of a word's behaviour. It does, however, present challenges.

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.

4.2 From collocation to Word sketches

One way of summarizing the data is to list the words that are found in close proximity of the word that is the subject of analysis, with a frequency far greater than chance; its collocations (Atkins & Rundell, 2008). The subfield of collocation statistics began with a paper by Church and Hanks (1989) who proposed a measure called Mutual Information (MI), from Information Theory, as an automatic way of finding a word's collocations: their thesis is that pairs of words with high mutual information for each other will usually be collocations. The approach generated a good deal of interest among lexicographers, and many corpus tools now provide functionality for identifying salient collocates, along these lines.[2]

One flaw of the original work is that MI emphasises rare words (and an ad hoc frequency threshold has to be imposed or the list would be dominated by very rare items). This problem can be solved by changing the statistic, and a number of proposals have been made. A range of proposals are evaluated in Evert and Krenn (2001) (though the evaluation is from a linguist's rather than a lexicographer's perspective). Statistics for measuring collocation, in addition to MI, include MI3, the log likelihood ratio, and the Dice coefficient; for a full account see Manning and Schütze (1999, chapter 5). Another, more recently proposed collocation statistic is logDice (Rychly, 2008).

Tables 1 to 4 below, each containing the top fifteen collocate candidates of the verb *save* in the OEC corpus, in the window of five tokens to the left and five tokens to the right, ordered according to MI, MI3, log likelihood, and logDice scores respectively, offer a good demonstration of the differences between different statistics. Collocate candidates offered by

MI are very rare, and not at all useful to lexicographers. Better collocate candidates, many of them the same, are offered by MI3 and log likelihood, however in this case very frequent functional words dominate the list. Even more useful candidate collocates are provided by logDice, from which the lexicographer can already get an idea about a few meanings of the verb *save*, for example 'use less of or invest' (*money*, *million*), 'prevent from harm' (*life)*, and 'store' (*file*). Collocation can thus be used not only to describe word meanings (Sinclair, 2004), but also to distinguish between them (see also Hoey, 2005). A list of collocates, representing an automatic summary of the corpus data, is therefore very useful for the lexicographer.

<<Table 1>>

<<Table 2>>

<<Table 3>>

<<Table 4>>

As shown in tables above, collocates are normally provided in the form of a list. Another way of displaying collocates, available in the COBUILD tools and WordSmith Tools, is called 'picture' (see Fig. 4) and lists collocates by frequency or by score of whichever statistic measure is used,[3] in each position between the selected span. The information in the Picture needs to be read vertically and not horizontally. Drawbacks of this display are that it gives the user a lot of information to wade through, and fails to merge information about the same word occurring in different positions.

<<Fig. 4>>

Word sketches

Collocation-finding as described above is grammatically blind. It considers only proximity. However, lexicographically interesting collocates are, in most cases, words occurring in a

particular grammatical relation to the node word. For example, the examination of concordance of the top collocates in Table 4 shows that a number of them occur as the object of the verb (e.g. *life, money, energy, file, planet*). In order to identify grammatical relations between words, the corpus has to be parsed.

Corpus feature combining collocation and grammar are Sketch Engine's 'word sketches'.[4] Word sketches are defined as "one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour" (Kilgarriff et al., 2004:105). Fig. 5 shows the word sketch for the verb *caress* in the ukWaC corpus (Ferraresi et al., 2008), which offers the lexicographer the most salient collocates that occur as the object, subject, modifier, or in the and/or relation of *caress* respectively.

<<Fig. 5>>

Word sketches were first used for the Macmillan English Dictionary (Rundell, 2002; Kilgarriff and Rundell, 2002). Atkins and Rundell (2008) see word sketches as a type of lexical profiling, which has become the preferred starting point for lexicographers when analysing complex headwords.

For word sketches to be built, the system must be told what the grammatical relations are for the language, and where in the corpus they are instantiated. There are two ways to do this. The input corpus may already be parsed, with grammatical relations given in the input corpus. Such a corpus is occasionally available.

The other way is to define the grammatical relations, and parse the corpus, within the tool. To do this, the input corpus must be POS tagged. Then each grammatical relation is defined as a regular expression over POS tags, using CQL. The CQL expressions are used to parse the corpus, giving a database of tuples such as <*subject, carress, breeze,* 14566778> where *subject* is a grammatical relation holding between the verb *caress* and the noun *breeze* at corpus reference point (for *caress*) 14566778. From the tuples database, word sketches are

generated at run-time. Parsing is done at compile time, and the results are stored, so users need not wait. The accuracy of the process is discussed and evaluated in Kilgarriff et al. (2010).

A list of collocates is sometimes directly transferred, by the lexicographer, from the corpus tool to the dictionary entry, as is demonstrated by Fig. 6 (Macmillan English Dictionary Online) where the box "Collocations: result" lists verbs that take *result,* in dictionary sense 3, as an object, as identified within the Sketch Engine.

<<Fig. 6>>

Thesaurus

The thesaurus feature provides a list of "nearest neighbours" (Kilgarriff et al., 2004:113) for the word. Nearest neighbours are the words that 'share most collocates' with their node word: if we have encountered <*subject, caress, breeze*> and <*subject, caress, wind*> then *breeze* and *wind* share a collocate: the process of generating the thesaurus is one of finding, for each word, which other words it shares collocates with (and weighting the shared items, see Rychly and Kilgarriff, 2007). The thesaurus provides a lexicographer with a list of potential (near-)synonyms (and, in some cases, antonyms). For example, the thesaurus output of the 10 nearest neighbours for the adjective *handsome* (1578 occurrences in the BNC), shown in Table 5, contains several synonym candidates, such as *good-looking, beautiful, pretty, lovely,* and *attractive.*

<<Table 5>>

Sketchdiffs

Sketch differences' or 'sketchdiffs' compare word sketches for the two words, showing the collocations that they have in common and those that they do not. Fig. 7 shows the sketch difference for adjectives *handsome* and *attractive* in ukWaC. Collocates *particularly, quite,*

extremely, *so*, *very*, *really* and *as* (highlighted in shades of red in the Sketch Engine) are more typical modifiers of *attractive*, *strikingly* and *devastatingly* (highlighted in green in the Sketch Engine) are more typical of *handsome*, while the remaining collocates in this relation show similar salience with both adjectives.

<<Fig. 7>>

The thesaurus and sketchdiff are linked. Clicking on a lemma in a thesaurus entry automatically opens the sketch difference comparing the original lemma with the one found in the thesaurus entry. Thesaurus and sketchdiffs were used extensively in compiling the *Oxford Learner's Thesaurus – a dictionary of synonyms* (Lea, 2008).

4.3 Good Dictionary EXamples (GDEX)

Good dictionary examples are hard to find; lexicographers have often invented, rather than found them, but that runs the risk of accidentally failing to provide a natural context for the expression being illustrated (cf. Hanks, this volume). Sketch Engine's GDEX attempts to automatically sort the sentences in a concordance according to how likely they are to be good dictionary examples (Kilgarriff et al., 2008). GDEX operates as an option for sorting a concordance: when it is on, the 'best' examples will be the ones that the user sees first, at the top of the concordance. GDEX scores sentences using heuristics for readability and informativeness. Readability heuristics include sentence length and average word length, and penalise sentences with infrequent words, more than one or two non-a-z characters, or anaphora. Informativeness heuristics include favouring sentences containing words that are frequently found in the vicinity of the expression that the concordance is for: it is likely that they are typical collocates for the expression. GDEX was first used in the preparation of an electronic version of *Macmillan English Dictionary*, 2nd edition, 2007.

GDEX was designed for English, so several heuristics are specific to the English language or were included with the needs of specific group of dictionary users in mind, i.e.

advanced learners of English. The usefulness of GDEX for other languages is thus limited. This has been confirmed by the experience with it when devising a new lexical database of Slovene in the "Communication in Slovene" project (www.slovenscina.eu), where the examples offered first by GDEX were rarely useful to lexicographers. Infrastructure for customising GDEX has recently been completed, and Slovene and other GDEXes are currently in development.

4.4 Why we still need lexicographers

No matter how many features help summarise the data, the lexicographer still needs to critically review the summary to determine the meaning of the word. Concordances should always be available to check the validity of results: there are many stages in the process where anomalies and errors might have arisen, from the source data, or in its preparation or lemmatisation or parsing. It needs to be easy for the lexicographer to check the data underlying an analysis, for any case where the analysis does not immediately tally with their intuitions.

One recurring area of difficulty, in all the languages for which we have been involved in lexicography – two recent examples being Polish and Estonian - is participles/gerunds. In English, most -ed forms can be verb past tense or past participle, or adjectival, and –ing forms can be verbal, adjective or gerunds, and comparable processes apply for most European languages. In theory, one might be able to distinguish the form (verbal participle) from the function (verbal, adjectival or nominal) but the theory still leaves the lexicographer with a judgement to make: should the –ing form get a noun entry, should the –ed form get an adjective entry? The analysis software is stuck with the same quandary: where we encounter an –ing form, should we treat it as part of the verb lemma or as an adjective, or as a noun. The problem has two parts: some syntactic contexts unambiguously reveal the function (*The painting is beautiful; he was painting the wall*) but many do not (*I like painting; the painting*

school) but this is only the first problem. The second problem is that some gerunds and participial adjectives are lexicalised, deserving their own entry in the dictionary, and others are not: thus we can have *the manoeuvring is beautiful* and there is no question that *manoeuvring* is functioning as a noun, but there is also no question that it is not lexicalised and does not need its own dictionary entry. The upshot is that many word sketches contain verb lemmas which are there misleadingly, because they are the result of lemmatisation of adjectival participles and gerunds, which should have been treated as adjective and noun lemmas in their own right.

5 Developing corpus tools to meet lexicographers' needs

Lexicographers are demanding corpus users, who get to understand the potential of corpora well and expect a wide range of features. Initially, not a great deal of thought was given to the actual look and user-friendliness of the interface – functionality and speed were more important. But with regular use of corpus tools, more time has to be spent on devising interfaces that are friendly to the lexicographers who use them on a daily basis. Training lexicographers on how to analyze data is time-consuming already, and a user-friendly interface helps them focus on analysis.

5.1 User-friendliness

A comparison of older tools with modern ones testifies to progress in user-friendliness. Conducting searches no longer requires typing in complex commands. Corpus tools have become more Google-like, where the users write the search term in the box, specify the search (often using a drop-down menu) if they want to, and promptly get what they want.

Another difference is in the use of colour. Black and white are no longer the only options, and modern tools use colour highlighting to aid navigation in the output (Fig. 8) and/or separate different types of information. For example, the sketchdiff uses green for

collocates more strongly associated with the first lemma, and red, for those more strongly associated with the second, with strength of colour indicating strength of the tendency.

<<Fig. 8>>

Some corpus tools offer graphical representations of numerical data. Graphical representation can often help lexicographers quickly identify usage-related information, for example an increase or decrease in the use of a word or phrase over a period of time (see Fig. 3), predominant use of the word in a certain domain, register, etc., typical use of the word in a specific form (e.g. when a noun occurs mainly in the plural form) and so forth.

Lexicographers have different preferences and use different equipment, such as computer screens of different sizes, so customizability is part of user-friendliness. An example of a basic customisable feature is adjustable font size. In the case of online corpus tools, font size can also be changed in the settings of the internet browser.

Many corpus tools also offer the option to change the Concordance output, in terms of how much data is displayed (e.g. the number of concordance lines per page, the amount of context shown), and which type of data is displayed, e.g. attributes of the searched item (word form, lemma, POS-tag, etc) and structure tags (document, paragraph, and sentence markers). A form of customisation requiring deeper understanding is control of the word sketches by changing parameters such as the minimum frequency of the collocate in the corpus, or the maximum number of displayed items. The Sketch Engine also provides ‘more data’ and ‘less data’ buttons to make the word sketches bigger or smaller.

Recent developments relating to character sets have been a great boon for corpus developers and lexicographers. Not so long ago, the rendition of the character set for each new language, particularly non-Latin ones, would have made a very large project each time. Now, with the Unicode standards and associated developments in character encoding methods, operating systems and browsers, these problems are largely solved, and well-

engineered modern corpus tools can work with any of the world's writing systems with very little extra effort. The Sketch Engine correctly displays corpora for Arabic, Chinese, Greek, Hindi, Japanese, Korean, Russian, Thai and Vietnamese, amongst others.

A related issue is the interface language. Chinese lexicographers working on Chinese, or Danes working on Danish, will not want an English-language interface. This has doubtless contributed to various institutions developing their own tools. The Sketch Engine is localisable, and currently the interface is available in Chinese, Czech, English, French and Irish.

5.2 Integration of features

Because of an increasing number of features offered by corpus tools, it is useful and time-saving if the features are integrated. The lexicographer looking at a list of collocates is likely to want to check the concordance lines of the collocate(s). If the collocation and the concordance features are integrated, the user can move between the two by mouse-click.

Another type of time-saving technique that could help lexicographers in the future would be to combine two features into one. An example of this can be found in the online tool for Gigafida, a 1.15-billion-word corpus of Slovene (which targets lay users and not lexicographers), where the Filters, which are offered in the menu to the left of the concordance output (see Fig. 9) and enable the user to filter concordance lines by basic forms, text type, source, and other categories, also provide frequency information for each available category in the filter (filter categories with zero concordance lines are not shown), ordering categories by frequency.

<<Fig. 9>>

5.3 Integration of tools

A corpus tool is not the only piece of software a lexicographer needs to master. There is always at least one other tool, the dictionary-writing system (cf. Adel, this volume). Lexicographic work often involves transferring corpus data to the dictionary database, and time and effort can be saved if the transfer is efficient. Copy-and-paste is possible in some cases, but often the information needs to be in a specific format (normally XML) for the dictionary-writing system to read it. This issue is addressed by the Sketch Engine's 'TickBox Lexicography'.

TickBox Lexicography (TBL) allows lexicographers to select collocates from the Word Sketch, select examples of collocates from a list of (good) candidates, and export the selected examples into the dictionary-writing system (see Fig. 10 and Fig. 11). An XML template, customised to the requirements of the dictionary being prepared, is needed for the data to be exported in the format compatible with the dictionary-writing system. The lexicographer does not need to think about XML: from their perspective, it is a simple matter of copy-and-paste.

<<Fig. 10>>

<<Fig. 11>>

Another option is to combine a corpus tool and a dictionary-writing system in a single program, so that lexicographers would use the same interface to search the corpus and write dictionary entries. Such software is already available, namely the TLex Dictionary Production System (Joffe & de Schryver, 2004), as reviewed in Abel (this volume).

5.4 Customisation

It often happens that a certain feature needs to be customised to the requirements of a particular dictionary project. A critical concern at the Institute for Dutch Lexicology (INL) was bibliographical references: in the ANW (a Dictionary of Contemporary Dutch, in

preparation), each example sentence is accompanied by its bibliographical details. They were available to the corpus system. However the time it took to type, or copy and paste, all those details into the appropriate fields in the dictionary-writing system was severely limiting the numbers of examples the lexicographers were using, and putting the whole project's schedule at risk. The Sketch Engine team was able to customise the TBL machinery to provide a 'special copy-and-paste' which automatically gathered together the bibliographic data for a sentence that the lexicographer had selected, and, on pasting, inserted the 'example', 'author', 'title' 'year' and 'publisher' into the appropriate fields of the dictionary-writing system.

Implementing a customised version of TBL does not require any changes to the corpus interface, but adding a new feature does. This has been the case with the Pattern Dictionary of English Verbs (Hanks & Pustejovsky, 2005; Hanks, 2008; Hanks, this volume) where lexicographers are using an enhanced version of the Sketch Engine, designed specially for the project to annotate concordance lines of the verb with the number of the associated pattern in the database entry (Fig. 12). In addition, the dictionary database is linked with the Sketch Engine so that the users can view all the concordance lines associated with a pattern with a single click.

<<Fig. 12>>

The relationship between the lexicographers working on a dictionary project, and the developers of the corpus tool used in the project is cyclical. Lexicographers benefit from the functionality of the corpus tools, and, since they are regular users of the tool and most of its features, provide feedback for the developers. This often results in further improvements to the tool, which again benefit lexicographers (as well as other users of the tool).

6 Conclusion

People writing dictionaries have a greater and more pressing need for a corpus than most other linguists, and have long been in the forefront of corpus development. From the Bank of

English corpus (used in the COBUILD project), to the BNC, the largest corpora were built for and used for lexicographic purposes (as well as for NLP purposes). Building large corpora is no longer problematic as many texts are readily available in electronic form on the internet. But exactly because corpora have got larger and larger, it has become more important that lexicographers have at their disposal corpus tools with summarisation features.

This chapter has shown that the functionality and user-friendliness of corpus tools have improved considerably since they were first used in dictionary projects. Corpus tools of today are faster and more diverse on the one hand, but easier to use on the other. Also, the needs of lexicographers have prompted the creation of features such as TickBox Lexicography, which ease the exporting of corpus information into the dictionary-writing system. Lexicographically-oriented features are also being used by linguists, teachers and others, which indicates that the distinction between lexicographic corpus tools and linguistic corpus tools is blurred.

There is, however, still more work to be done in terms of making corpus tools as useful to lexicographers as possible. This includes coming up with more features that bridge the gap between raw corpus data and the dictionary. One strategy is to establish a closer link between corpus tool and dictionary-writing system, with more features like TickBox Lexicography supporting seamless data transfer. Currently, most of the focus is on examples; definitions are written in the dictionary-writing system, which means the lexicographer may need to switch between corpus tool and dictionary-writing system quite often. Corpus tools of the future should perhaps offer a more complete solution, e.g. allowing the lexicographer to mark examples, devise a draft definition (in a pop-up window) and any other part of the meaning in the corpus tool, and only then export into the dictionary entry.

Corpora and associated software do more and more by way of summarising the information to be found about a word or phrase. A question worth asking then is: will corpus

tools reach a point where they act as dictionaries? The idea does not seem too far-fetched. There is already research showing that definitions of words can be extracted directly from corpora (Pearson, 1996; 1998). Also, there is already, in GDEX, a feature available that helps identify good dictionary examples. Nonetheless, as Rundell and Kilgarriff (in press) point out, providing the users with automatically extracted corpus data, rather in a traditional dictionary format, may pose problems for some types of users, for example language learners. The position we take is this: lexicographers are better at preparing brief, user friendly accounts of a word's meaning and behaviour than automatic tools – but they have not covered everything, as no dictionary covers all the new and obscure words, specialised uses, contextually appropriate collocations. Where a user wants to find something out, it is most convenient if they can find it in a dictionary; but if the dictionary does not meet their needs, then yes, they should turn to the corpus.

References

a. Corpora and corpus tools

Anthony, Laurence (2011) AntConc, version 3.2.2.1. Tokyo, Japan: Waseda University. Available from http://www.antlab.sci.waseda.ac.jp/.

Barlow, George Michael (2002) MonoConc Pro 2.2. Athelstan.

Barlow, George Michael (2009) MonoConc Easy. Athelstan.

British National Corpus (BNC). http://natcorp.ox.ac.uk.

Corpus.byu.edu. http://corpus.byu.edu.

Corpus de Referencia del Español Actual (CREA). http://corpus.rae.es/creanet.html.

Corpus Pattern Analysis (CPA). http://corpora.fi.muni.cz/cpa.

Corpus WorkBench, version 3.0.0. Stuttgart: IMS, University of Stuttgart. Available from http://cwb.sourceforge.net/download.php.

Gigafida corpus of Slovene. http://demo.gigafida.net

DeepDict Lexifier. http://gramtrans.com/deepdict.

KorpusDK. http://ordnet.dk/korpusdk.

Oxford English Corpus. http://oxforddictionaries.com/page/oec.

Scott, Mike (2008) *WordSmith Tools version 5*. Liverpool: Lexical Analysis Software.

Sketch Engine. http://the.sketchengine.co.uk.

Webcorp. http://www.webcorp.org.uk.

b. Literature

Atkins, Beryl T. Sue, and Michael Rundell (2008) *The Oxford Guide to Practical Lexicography*. Oxford: Oxford University Press.

Bick, Eckhard (2009) DeepDict - A Graphical Corpus-based Dictionary of Word Relations. *Proceedings of NODALIDA 2009. NEALT Proceedings Series Vol. 4*, 268-71. Tartu: Tartu University Library.

Christ, Oliver (1995) *The IMS corpus workbench technical manual, Technical report*. Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Church, Kenneth Ward, and Patrick Hanks (1989) Word association norms, mutual information and lexicography. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 76-83. Vancouver.

Clear, Jeremy (1987) Computing. In John McHardy Sinclair (ed.) *Looking up: An Account of the COBUILD Project in Lexical Computing*, 41-61. London: Collins ELT.

Davies, Mark (2009) The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics* 14(2): 159-90.

Evert, Stefan, and Brigitte Krenn (2001) Methods for the Qualitative Evaluation of Lexical Association Measures. *Proceedings of the 39th Annual Meeting of the Association for Computational* Linguistics, 188-95. Toulouse, France: ACL.

Ferraresi, Adriano, Eros Zanchetta, Marco Baroni, and Silvia Bernardini (2008) Introducing and evaluating ukWaC, a very large web-derived corpus of English. In Evert, Stefan, Adam Kilgarriff, and Serge Sharoff (eds.) *Proceedings of the 4th Web as Corpus Workshop (WAC-4) – Can we beat Google?* Marrakech, 1 June 2008.

Hanks, Patrick, and James Pustejovsky (2005) A Pattern Dictionary for Natural Language Processing. *Revue Francaise de linguistique appliquée* 10(2): 63-82.

Hanks, Patrick (2008) Mapping meaning onto use: a Pattern Dictionary of English Verbs. AACL 2008, Utah.

Hoey, Michael (2005) *Lexical priming.* Oxford: Routledge.

Joffe, David, and Gilles-Maurice de Schryver (2004) TshwaneLex – A State-of-the-Art Dictionary Compilation Program. In Williams, Geoffrey, and Sandra Vessier (eds.) *Proceedings of the 11th Euralex International Congress*, 99-104. Lorient, France: Université de Bretagne Sud.

Kehoe, Andrew, and Antoinette Renouf (2002) WebCorp: Applying the Web to Linguistics and Linguistics to the Web. *Proceedings of the World Wide Web Conference*, Honolulu, Hawaii. Available at: http://www2002.org/CDROM/poster/67/.

Kilgarriff, Adam, and Gregory Grefenstette (2003) Introduction to the Special Issue on Web as Corpus. *Computational Linguistics* 29(3): 333-48.

Kilgarriff, Adam, Miloš Husák, Katy McAdam, Michael Rundell, and Pavel Rychly (2008) GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In Bernal, Elisenda, and Janet DeCesaris (eds.) *Proceedings of the 13th EURALEX International Congress*, 425-32. Barcelona, Spain: Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra.

Kilgarriff, Adam, Vojtech Kovar, Simon Krek, Irena Srdanović, and Carole Tiberius (2010) A quantitative evaluation of word sketches. *Proceedings of the 14th EURALEX International Congress*, 372-79. Leeuwarden, the Netherlands.

Kilgarriff, Adam, and Michael Rundell (2002) Lexical profiling software and its lexicographic applications - a case study. In Braasch, Anna, and Claus Povlsen (eds.) *Proceedings of the 10th Euralex International Congress*, 807-18. Copenhagen: August.

Kilgarriff, Adam, Pavel Rychly, Pavel Smrz, and David Tugwell (2004) The Sketch Engine. In Williams, Geoffrey, and Sandra Vessier (eds.) *Proceedings of the 11th Euralex International Congress*, 105-16. Lorient, France: Université de Bretagne Sud.

Kosem, Iztok (2010). Designing a model for a corpus-driven dictionary of academic English. PhD thesis. Birmingham: Aston University.

Lea, Diana (2008) Making a Thesaurus for Learners of English. In Bernal, Elisenda, and Janet DeCesaris (eds.) *Proceedings of the 13th EURALEX International Congress*, 543-50. Barcelona, Spain: Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra.

Macmillan English Dictionary online. http://www.macmillandictionary.com.

Manning, Christopher D., and Hinrich Schütze (1999) *Foundations of Statistical Natural Language Processing*. MIT Press.

Pearson, Jennifer (1996) The Expression of Definitions in Specialised Texts: A Corpus-based Analysis. In Gellerstam, Martin (ed.) *Euralex '96 Proceedings*, 817-24. Gothenburg: Gothenburg University.

Pearson, Jennifer (1998) *Terms in Context (Studies in Corpus Linguistics)*. Amsterdam: John Benjamins Publishing Company.

Rundell, Michael (ed.) (2002) *Macmillan English Dictionary for Advanced Learners, First Edition*. London: Macmillan.

Rundell, Michael (ed.) (2007) *Macmillan English Dictionary for Advanced Learners, Second Edition*. London: Macmillan.

Rundell, Michael, and Adam Kilgarriff (in press) Automating the creation of dictionaries: where will it all end? In Meunier, Fanny, Sylvie de Cock, Gaëtanelle Gilquin, and Magali Paquot (eds.), *A Taste for Corpora. In honour of Sylviane Granger*. Amsterdam: John Benjamins.

Rychly, Pavel (2008) A lexicographer-friendly association score. In Sojka, P. & Horák, A. (eds.) *Proceedings of Second Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2008*, 6-9. Brno: Masaryk University.

Rychly, Pavel, and Adam Kilgarriff (2007) An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments). In *Proceedings of the*

45th Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic.

Scott, Mike (1997) PC Analysis of Key Words – and Key Key Words", *System* 25 (3): 1-13.

Scott, Mike, and Christopher Tribble (2006) *Textual Patterns: keyword and corpus analysis in language education*. Amsterdam: Benjamins.

Sinclair, John McHardy (1991) *Corpus, concordance, collocation.* Oxford: Oxford University Press.

Sinclair, John McHardy (2004) *Trust the text.* London: Routledge.

Footnotes

1. An online version of the tool is freely accessible, with limitations on searches (e.g. the maximum number of displayed hits is 1000).

2. In our terminology, a *collocation* comprises *node word + collocate(s)*, in particular grammatical relations.

3. WordSmith Tools lists collocates in the picture view by frequency only.

4. A similar feature is also provided by the DeepDict Lexifier tool (Bick, 2009).

Tables

Table 1: Top 15 collocates of the verb *save* (ordered by MI score).

lemma	freq	MI
BuyerZone.com	7	13.192
ac);	5	13.192
count-prescription	5	13.192
Christ-A-Thon	7	13.192
Teldar	6	12.607
Re:What	26	12.535
Redjeson	5	12.514
INFOPACKETS30	3	12.455
other-I	4	12.385
SetInfo	4	12.385
Ctrl-W	9	12.362
God	18	12.233
Walnuttree	3	12.192
Hausteen	5	12.192
MWhs	3	12.192

Table 2: Top 15 collocates of the verb *save* (ordered by MI3 score).

lemma	freq	MI3
to	99846	37.289
life	27606	36.975
.	102829	36.652
money	19901	36.513
the	106241	36.388
,	86327	35.686
be	70859	35.253
and	62030	35.218
from	28399	34.437
a	47129	34.139
of	41271	33.380
have	29869	33.213
you	20610	33.021
that	29260	33.012
for	25291	32.901

Table 3: Top 15 collocates of the verb *save* (ordered by log likelihood score).

lemma	freq	log likelihood
to	99846	417.952.128
.	102829	333.836.913
the	106241	297.431.943
life	27606	234.592.446
,	86327	222.779.392
and	62030	192.235.461
be	70859	190.628.164
money	19901	181.861.449
from	28399	139.301.252
a	47129	126.211.751
have	29869	92.837.927
of	41271	90.602.606
you	20610	86.952.618
that	29260	85.631.777
for	25291	83.634.156

Table 4: Top 15 collocates of the verb *save* (ordered by logDice score).

lemma	freq	logDice
money	19901	9.344
life	27606	9.048
save	2976	7.518
energy	2648	7.368
million	4742	7.168
dollar	1847	7.158
file	2147	7.139
try	6380	7.108
$	6193	7.074
could	11904	7.054
effort	2844	7.049
	2583	7.010
retirement	1181	6.940
planet	1194	6.906
thousand	1894	6.891

Table 5: 10 nearest neighbours for *handsome* offered by Thesaurus.

lemma	**similarity score**
good-looking	0.271
elegant	0.238
charming	0.236
beautiful	0.233
pretty	0.233
tall	0.218
lovely	0.202
attractive	0.202
clever	0.197
slim	0.197

Figures

<<Fig. 1: The KWIC format of concordance output.>>

<<Fig. 2: Concordances from Fig. 1, sorted by the first word to the right.>>

<<Fig. 3. Frequency distribution of lemma *random* in blogs subcorpus of the OEC.>>

<<Fig. 4: The Picture view for *local* in WordSmith Tools. (Source: Corpus of Academic Journal Articles; Kosem, 2010)>>

<<Fig. 5: Word sketch of the verb *caress* (in the ukWaC corpus).>>

<<Fig. 6: entry *result*, collocations under sense 3. (Source: Macmillan English Dictionary online)>>

<<Fig. 7: Sketch difference for adjectives *handsome* and *attractive*.>>

<<Fig. 8: Marking the concordance line that the user is examining (Source: KorpusDK).>>

<<Fig. 9: The Filters in the tool for accessing the Gigafida corpus.>>

<<Fig. 10: Selecting collocates with TickBox Lexicography.>>

<<Fig. 11: Selecting examples of the collocates, selected in TBL, for export.>>

<<Fig. 12: Annotated concordance lines of the verb *abate* (accessed via Corpus Pattern Analysis extension).>>