Peter Uhrig and Thomas Proisl[1]

Less hay, more needles – using dependency-annotated corpora to provide lexicographers with more accurate lists of collocation candidates

Abstract

Collocations in dictionaries are often based on automatically extracted candidate lists from large text corpora filtered by a lexicographer. The present paper discusses the two currently most popular approaches to the extraction process, the traditional window-based and the more recent Part-of-Speech-pattern approach. As an improvement on current practices, we suggest to use a third approach to collocation candidate extraction based on dependency-annotated corpora. All three methods are evaluated against an existing collocations dictionary, revealing that the dependency-based approach can in general significantly improve the quality of the candidate lists. Finally, a tool that allows lexicographers to use dependency-annotated versions of their own corpora by means of a simple web interface will be presented.

1. Collocation and lexicography

It is probably unnecessary to stress the importance of collocation to lexicography – particularly to bilingual foreign language lexicography and to learner lexicography – in a journal that recently devoted almost an entire volume to “Collocations in European lexicography and dictionary research” (*Lexicographica* 24, 2008). Ever since the focus in foreign language pedagogy shifted from teaching isolated words to teaching words in their “natu-

[1] The order of authors is arbitrary.

DOI 10.1515/lexi.2012–0009

ral environment" in the 1980s, the treatment of collocations in dictionaries has been widely discussed.[2] The publication of COBUILD1 (1987) marks the introduction of computationally extracted and manually verified collocation data into lexicography, a process that has since gained tremendous popularity and can be considered mainstream. Today, all major learner's dictionaries of English devote considerable attention to collocation (see for instance Herbst/Mittmann 2008 or Götz-Votteler/Herbst 2009 for a survey) and there are specialised collocations dictionaries such as the *BBI Combinatory Dictionary of English* (first edition 1986, third edition 2010), the *Oxford Collocations Dictionary for students of English* (first edition 2002, second edition 2009; henceforth OCD1/OCD2) and – the most recent publication – the *Macmillan Collocations Dictionary for Learners of English* (2010; henceforth MCD).

The present paper sets out to discuss current practices of and potential improvements on the computational extraction of collocations, the software and methodology for which have evolved to rather advanced levels. In this introductory section, we will first have to briefly discuss the theoretical status and various notions of collocation together with its relation to lexicography. The second section will give an overview of established techniques for collocation candidate extraction from corpora, in section 3 we will present a more sophisticated approach to the problem based on full syntactic parses and the resulting dependency structures. We will compare and evaluate all approaches in section 4, showing that the dependency-based approach is superior to other approaches. Section 5 briefly presents *Treebank.info*, a freely available web interface implementing dependency-based collocation candidate extraction, and discusses consequences for lexicographic work.

1.1 Theoretical notions of collocation

When Hausmann stated in 2003 (published as Hausmann 2004) that there is a "terminological war" about the term collocation and claimed that many computational and corpus linguists were not even aware of it, he was certainly exaggerating. Nonetheless it is necessary to take a look at the two major uses of the term.[3]

In the tradition of Firth ("you shall know a word by the company it keeps" (Firth 1957/1968, 179)), Sinclair defines the term collocation as "the occurrence of two or more words within a short space of each other in a text" (Sinclair 1991, 170). In computational implementations, the "short space" often corresponds to a so-called "window" of several orthographic words to the left and right (often 5; see discussion in 2.5), so we shall use the term window-based approach for such extraction methods. In this very general sense of collocation, any of the combinations given in Kjellmer's (1994) dictionary[4] can be regard-

[2] In Germany, Hausmann's (1984; 1985) publications can probably be seen as the starting point for the discussion, even though Hausmann himself is careful to show that the concept and the term had been widely used before and not only by researchers in British contextualism (see Hausmann 2008, 5–6).

[3] We shall not cover here the text-linguistic use of the term as defined by Halliday/Hasan (1976, 287) due to its limited relevance to lexicography.

[4] Kjellmer's dictionary was not listed among the collocations dictionaries above since it is not aimed at foreign language learners but at researchers and was created without manual intervention.

ed as a collocation, for instance *hotel at*, *a downtown hotel*, *at her hotel*, *left the hotel*. However, Sinclair further restricts the definition in order to exclude some such uses and states that collocation "in its purest sense [...] recognizes only the lexical co-occurrence of words" (Sinclair 1991, 170). He then goes on to state that the concept "is often related to measures of statistical significance" (Sinclair 1991, 170). It is this view of collocations that is most widely used by corpus linguists and computational linguists, but it is also used in lexicography (for instance in the selection of examples in the first edition of the Cobuild dictionary). In his comparison of the various uses of the term collocation, Herbst cites "sandy beaches" and "sell a house" as typical examples of this position (Herbst 1996, 384) since they are statistically significantly associated in corpora even though they are free combinations semantically.[5]

The second approach to collocation we shall cover here is the one advocated by Hausmann (1979; 1984; 1985; 2004), which is inspired by the problems foreign learners of a language face when trying to produce idiomatic text and their requirements on dictionaries to provide them with the necessary information. Hausmann's model limits the concept of collocation to a relationship between exactly two items, one of which he calls base ("Basis"), the other collocate[6] ("Kollokator"). The base is a semantically autonomous word such as *table*, the collocate a word that shows a certain affinity (Hausmann 1984, 398) to occur with the base and that can often only be interpreted semantically in the context of the respective base, such as *lay* in the context of *table*.[7] According to Hausmann, the learner starts off with the base because he/she wants to make a statement about it and then needs the right collocate for the respective base. The distinction is tied to word class, so usually nouns are bases while verbs and adjectives are collocates of these nominal bases.[8] Accordingly, base and collocate are usually connected by some sort of syntactic relation. This is also echoed in Bartsch's working definition:

> Collocations are lexically and/or pragmatically constrained recurrent co-occurrences of at least two lexical items which are in a direct syntactic relation with each other. (Bartsch 2004, 76)

Hausmann argues strongly for the inclusion of collocates in the dictionary entries of bases for production purposes since the learner can find them only there when he/she does not know them already.[9] Herbst cites *false teeth* and *artificial leg* as typical examples, where the learner has to know that *artificial teeth* and *false leg* are not the conventional wordings,

[5] Nonetheless, as Herbst (2011) shows, even some of these apparently free combinations must be learned since they represent conceptual units and there is no way to predict that the concept of a sandy beach is usually expressed in the form of a premodifying adjective plus a noun in English and not – as for instance in German – as a compound (*Sandstrand*, "sand beach").

[6] Lea (2007) uses the English term *collocator* instead.

[7] Example borrowed from Hausmann (2004, 309).

[8] Of course verbs and adjectives are also bases when it comes to their modification by adverbs.

[9] While this is highly plausible from a lexicographic perspective, one may argue that cognitively many collocations are stored as conceptual units as a whole and not analysed into base and collocate in the same way (see Herbst 2011 for a brief discussion). Tarp (2008: 253) however also challenges Hausmann's position from a lexicographic point of view and argues that the collocation should also be given in the collocate entry even for production purposes.

and indeed research on learner language shows that learners produce significantly more errors on such collocations than on free combinations (Nesselhauf 2005).[10]

Many researchers actually make a terminological distinction that roughly corresponds to the two positions outlined here, so there is the semantically motivated distinction between open collocation[11] and restricted collocation (Cowie 1981), between cooccurrence and collocation (Evert 2005, 17) or between collocation candidates and collocations (Heid 1998, 301).[12] The latter distinction will be used here, since for lexicographic purposes, the idea is that collocation candidates are extracted from a corpus and then filtered by a lexicographer as to whether they are actual collocations that merit inclusion in a dictionary.[13] It has to be made clear, though, that it is very unlikely that all collocations in Hausmann's sense can be identified by such a method given that frequency is not a necessary criterion.

We shall conclude this section with a very strong claim made by Hausmann with regard to the "war" on collocation and will discuss in the next section, to what extent it can be justified for lexicographic applications:

> Der basisbezogene Kollokationsbegriff ist der engere, der merkmalreichere, der elaboriertere, der genauere, der funktionalisiertere, der anwendungsbezogenere, folglich der unverzichtbarere. (Hausmann 2004, 321)[14]

1.2 Collocation in practice

Hausmann made his statement cited above after the publication of OCD1 in 2002 and asserts that the fact that the dictionary follows the practice of distinguishing between bases and collocates (without using the terms in the front matter)[15] is a clear indicator that for purposes of foreign language lexicography and pedagogy, the base-collocate-notion of collocation is the superior one. A further argument in his favour is that many of the collocations of the statistically significant type are relatively straightforward, as also noted by Götz-Votteler/Herbst in an analysis of collocations in learner's dictionaries, where they state that "the question remains as to whether it is actually necessary to include combinations such as *a good*, *bad*, *right*, *wrong decision* or also *regret a decision*, as the semantic pos-

[10] However, it has to be noted that Nesselhauf also found that the availability of a dictionary in an exam situation had no significant effect on the accuracy of collocation use. She attributes this fact to a lack of awareness of collocational problems (2005, 238).

[11] Open collocations in Cowie's sense correspond probably more to co-creations in Hausmann's sense.

[12] See also the thorough discussion in Siepmann (2005), who uses the terms "frequency-based approach" and "semantically-based approach" (Siepmann 2005, 411).

[13] This was the approach taken by the compilers of OCD1, where the lexicographers acted as filters: "It was necessary, for each entry, somehow to draw a line between what should be included and what should not. This line could not be based solely on frequency, nor on statistical significance, but was informed by both of these. But it was informed also by editorial judgement about what would be useful to a learner consulting the dictionary." (Lea 2007, 267)

[14] The base-related notion of collocation is the more narrow one, the more feature-rich one, the more elaborated one, the more exact one, the more functionalised one, thus the more indispensable / essential one.

[15] Lea (2007, 268) however makes it very clear that this is the concept of collocation used in the dictionary.

sibility of combining these words is neither unexpected nor beyond an advanced learner's linguistic knowledge" (Götz-Votteler/Herbst 2009, 53).[16] However, OCD2 also lists these collocations,[17] as well as *sandy beach* and *sell a house* mentioned above, so Hausmann's "victory" could be regarded as only partial.

The latest addition to the market of collocations dictionaries, MCD, departs from the practice of OCD1/2 in two ways relevant here.[18] First, it also contains adjective-noun collocations in some, but by far not in all adjective entries, so it appears as if the compilers made a distinction between such collocations with a nominal and an adjectival base (and similarly for verb-noun collocations). For instance, there is an entry for *disadvantaged*, which contains the collocate *household*, but *household* is not listed in the dictionary as a base. There are also *economic* or *financial*, where it is just as likely that the learner knows the adjective and needs the corresponding noun (such as *growth* or *institution* respectively) as the other way round, so the collocations are listed in both the noun and the adjective entry. Such an approach appears to be very user-friendly and presents a strong case against the strict word-class-specific approach advocated by Hausmann.

Secondly, MCD is quite selective as to which headwords it includes.[19] It thus excludes words "like *house*, *buy* and *good*, which have no strong collocates at all: just about any word can be (and does) combine [sic] with words like these, as long as the combination makes sense" (Rundell 2011). The argument is in line with the view taken by Götz-Votteler/Herbst cited above and the approach is thus somewhat closer to Hausmann's position than OCD2's, although there are of course some combinations (such as *terraced house* or *semi-detached house*) that would qualify as collocations. Whether the omission of entries for some of these frequent words will be accepted by the users of the dictionary thus remains to be seen.[20]

2. Extracting collocation candidates from corpora – the state of the art

Today, many pieces of corpus processing software offer a method for the extraction of collocation candidates by means of statistical tests. To give a basic example, *AntConc* (Anthony 2007) can process plain text corpora and produce a list of collocation candidates for a given word. Figure 1 is an example of such a list for *time* in the Brown Corpus (Francis 1965).

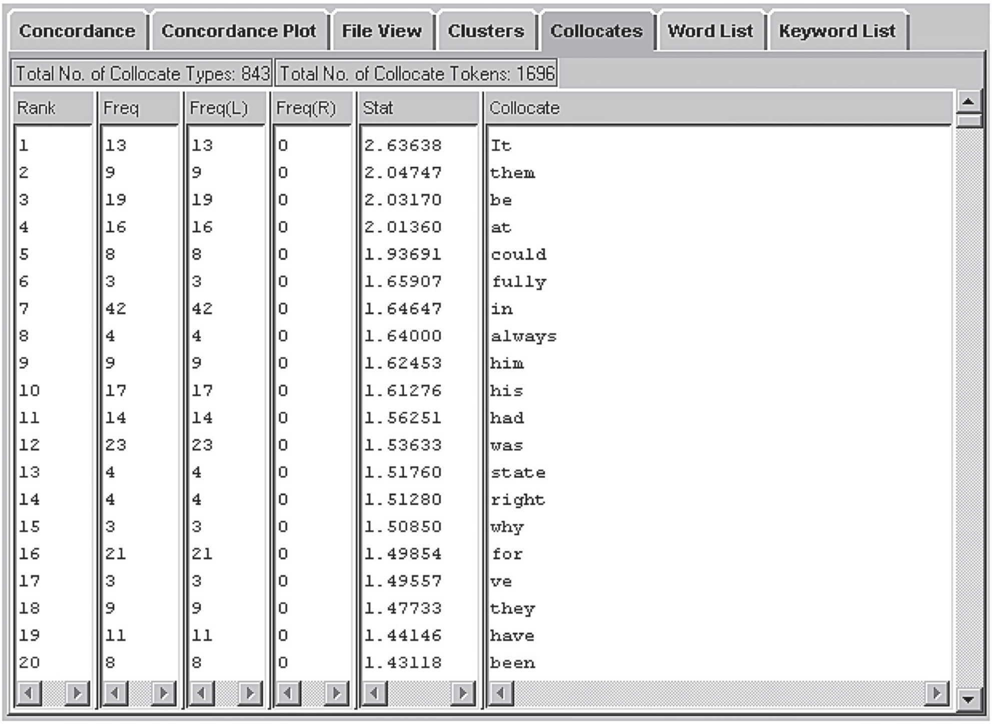
[16] These are open collocations in Cowie's sense, who also argues against their inclusion in a collocations dictionary.

[17] For *decision*, the collocate *regret* was added in OCD2; all others were also included in OCD1.

[18] There are of course others, such as the definitions and the labels for semantic groups, both of which indicate a target audience at a less proficient level than is necessary to fully benefit from OCD2.

[19] With the advertised headword count half of what OCD2 advertises (4,500 vs. 9,000 headwords), the reduction is quite substantial but is probably necessary due to the often more detailed entries; see also footnote 18.

[20] There is no entry for *water*, for instance, but the entry in OCD2 sports a large array of potentially not entirely uninteresting collocates.



Total No. of Collocate Types: 843 Total No. of Collocate Tokens: 1696

Rank	Freq	Freq(L)	Freq(R)	Stat	Collocate
1	13	13	0	2.63638	It
2	9	9	0	2.04747	them
3	19	19	0	2.03170	be
4	16	16	0	2.01360	at
5	8	8	0	1.93691	could
6	3	3	0	1.65907	fully
7	42	42	0	1.64647	in
8	4	4	0	1.64000	always
9	9	9	0	1.62453	him
10	17	17	0	1.61276	his
11	14	14	0	1.56251	had
12	23	23	0	1.53633	was
13	4	4	0	1.51760	state
14	4	4	0	1.51280	right
15	3	3	0	1.50850	why
16	21	21	0	1.49854	for
17	3	3	0	1.49557	ve
18	9	9	0	1.47733	they
19	11	11	0	1.44146	have
20	8	8	0	1.43118	been

Figure 1: Collocates for *time* in the Brown Corpus, sorted by t-score, window size 5, no frequency threshold, extracted with the help of *AntConc* (Anthony 2007)

As we can see, the list produced by such a simple method is far from satisfactory for the lexicographic treatment of collocations since it contains many irrelevant function words, contains various forms of the same lemma (*have*, *ve*, *had* or *be*, *was*, *been*), and does not group the results according to their grammatical category or their grammatical relation to the search item.

We shall briefly review common practices for such problems related to collocation extraction in the remainder of this section.

2.1 Part-of-Speech tagging

Using Part-of-Speech (PoS) tagging as, for instance, suggested by Church/Hanks (1990) considerably improves the usability of collocation extraction since it is possible to limit the results of such a process to bases and collocates of a certain word class and exclude others (usually the closed classes of function words). A tagger also solves the frequent problem of inter-word class homonymy.[21] In the example given above, one could for instance limit

[21] The impressive figures published in evaluations of automatic PoS taggers (“The error rate of state-of-the-art taggers is between 2 and 5%” (Schmid 2008, 547)) however obscure the fact that

the search to noun uses of *time* and restrict the collocates to verbs in order to find some lexicographically relevant combinations such as *elapse* or *pass*. We would argue that for the relatively coarse distinctions usually made in collocation extraction, the choice of a tagset is secondary, so the various applications of the CLAWS tagset (mainly used by CLAWS, the Part-of-Speech tagger, see for instance Leech et al. (1994)) or the Penn Treebank tagset (Marcus et al. 1993; used by a wide range of taggers and parsers, e.g. the Stanford suite (<http://nlp.stanford.edu>) or the TreeTagger (Schmid 1994)) should be equally well-suited for the task. Whatever system is used, pre-processing and suitable collocation extraction software are needed in order to use the tags for the extraction process.

2.2 Lemmatization

Given that the lemma is the basic unit of lexicographic description for most dictionaries (see for instance Crystal's definition of the term as "item which occurs at the beginning of a dictionary entry" or "headword" (2008, 273)), automatic lemmatization, i.e. procedures to relate word forms occurring in texts to their lemma forms, makes the job of a lexicographer easier since various word forms (such as *be*, *was*, *been* in the example given above) are subsumed under their base form. There is however serious criticism of such a simplification from the adherents of corpus-driven methods, for instance by John Sinclair.

> It is now possible to compare the usage patterns of, for example, all the forms of a verb, and from this to conclude that they are often very different one from another. There is a good case for arguing that each distinct form is potentially a unique lexical unit, and that forms should only be conflated into lemmas when their environments show a certain amount and type of similarity. (Sinclair 1991, 8)

Taking up this point, Tognini-Bonelli (2001, 92–98) cites evidence from two corpora to show that the most significant co-occurrences with the forms *facing* and *faced* differ substantially: "One glance at the collocational profiles [...] dispels any possible illusion that inflected forms are grammatical variations of a certain base form, but broadly share the same meaning of the base form and have a similar behaviour" (Tognini-Bonelli 2001, 94).[22]

Given constraints on time and space, not all lexicographers may agree with Tognini-Bonelli for all their projects, so it is important for computer systems for collocation candidate extraction that they give the user a choice of whether lemmatization shall be used or not.[23]

taggers still produce a considerable number of errors on homonymous forms and very rarely are wrong on frequent words such as *the* or *of*. With a large enough corpus, such effects can hopefully be neglected.

[22] Most lemmatizers would group *facing* and *faced* together under the lemma *face* (verb), but not *education* and *educate* as Tognini-Bonelli claims was done in a study by Stubbs (see Tognini-Bonelli 2001, 92f).

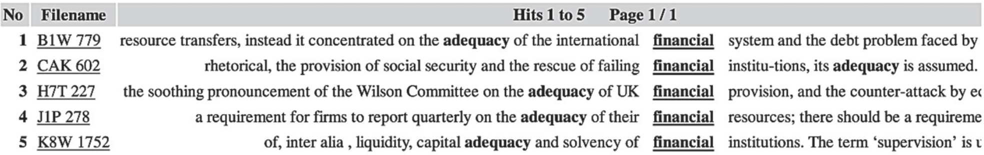
[23] While this position is certainly valid for languages such as English, German or French, it appears that the notion of lemma is much less straightforward for other languages; see for instance Knowles/Zuraidah (2004) for evidence from Arabic and Malay.

2.3 Association measures

The degree to which a pair of two items is associated is usually determined with the help of an association measure. There is a large body of literature on the subject and a wide range of such measures has been proposed, some based on tests of statistical significance, some from an information-theoretical point of view, some as heuristic variations on these.[24] In the present paper, we only consider a small sample of commonly used association measures in our comparison. These are Mututal Information (MI) and t-score (see Church/Hanks 1990), log-likelihood (Dunning 1993), Dice (Smadja 1993) and MI3 (Daille 1994). A thorough account of association measures with all their specificities is given in Evert (2005), a more concise summary that is possibly more accessible to linguists in Evert (2008).

2.4 Thresholds

In principle, there are three common ways to improve collocation candidate lists by means of a threshold. The simplest one is to exclude combinations that occur less than n times together, i.e. to impose a co-frequency threshold. For the BNC with its 100 million words a minimum of 4 (as in the *Sketch Engine*[25], Kilgarriff et al. 2004) or 5 (as in *BNCweb*) appears to be a reasonable default, although this will still let more frequently used proper names through. In a window-based approach, frequent items may still occur 5 times in the same 5-word window by chance, so while there is no syntactic and semantic relation between *financial* and *adequacy* in the following corpus lines, they amount to a total of 5 and thus will be displayed with the default threshold in *BNCweb*:



No	**Filename**	**Hits 1 to 5 Page 1 / 1**		
1	B1W 779	resource transfers, instead it concentrated on the **adequacy** of the international	**financial**	system and the debt problem faced by
2	CAK 602	rhetorical, the provision of social security and the rescue of failing	**financial**	institu-tions, its **adequacy** is assumed.
3	H7T 227	the soothing pronouncement of the Wilson Committee on the **adequacy** of UK	**financial**	provision, and the counter-attack by ec
4	J1P 278	a requirement for firms to report quarterly on the **adequacy** of their	**financial**	resources; there should be a requireme
5	K8W 1752	of, inter alia , liquidity, capital **adequacy** and solvency of	**financial**	institutions. The term ‘supervision’ is u

Figure 2: Concordance display of the collocation candidate *financial* + *adequacy* in *BNCweb*

On the other hand, setting the threshold higher in order to avoid such items is counter-productive, too, as discussed by the editor of OCD1:

[24] The choice is made even more complicated by the fact that it is possible to combine association measures in order to counter idiosyncrasies of the individual measures. For instance, Lehman/ Schneider (2011) use O/E (a variant of MI) and then test for statistical significance using t-score. See however the quote by Evert in 2.4 on the peculiarity of t-score as a filter of statistical significance.

[25] Interestingly, the *Sketch Engine* also offers an “automatic” setting for minimum frequency. While the documentation claims that it is “a function of corpus size” (<http://trac.sketchengine.co.uk/wiki/SkE/Methods/methods>, retrieved 10 May 2012), the actual implementation seems not to enforce any limit on the BNC.

> The first point we observed was that some of the strongest collocations – including some of those that spring most readily to mind when trying to explain the concept of collocation to someone to whom it is unfamiliar – are actually pretty rare. For example, *auspicious occasion* occurs only 7 times in the 100 million word British National Corpus. Similarly, *cushy job* has only 7 citations; *rancid butter* – quoted by Thierry Fontanelle [sic] (along with the more frequent *sour milk* and *rotten eggs*; 1994, 42) – has 6, and *arrant nonsense* 5. (Lea 2007, 266)

It is also possible to set a threshold for the association score, but only for some association measures this will correspond to a level of statistical significance (e.g. not for Mutual Information or MI3). And even then, many tests for significance assume a random distribution that is certainly not met in a language corpus, so the significance level can only be seen as a rough approximation only comparable for the same item (see Hoffmann et al. 2008, 157f). One special case is t-score, where a significance level threshold results in a "built-in" frequency threshold:

> Unlike all other measures in this group, t-score sets an implicit frequency threshold: no pair type with $o \leq 22$ can achieve a significance of $pv = 10^{-6}$, regardless of its expected frequency. Even for the customary significance level of $pv = .01$, there is an implied frequency cutoff at $o = 5$. This unique property of t-score might explain its success for filtering out unwanted candidates in collocation extraction tasks (Church et al. 1991), where it has possibly worked more as a frequency filter than as a test of significance. (Evert 2005, 114)

Nonetheless thresholds of association measures can be used to shorten the lists from the bottom and exclude collocates that are rather repelled than attracted by the base, but it has to remain clear that any such threshold has the character of a heuristic and is thus not necessarily better than a simple frequency threshold.

With a large sample corpus, one may also impose a threshold on the number of texts in which the collocation candidate has to occur in order to be listed, which is basically a very simple measure of dispersion. Such a threshold can not only eliminate proper names used frequently in few texts (e.g. *Skeldale House* (6x in one text) or *Chesser House* (25x in three texts) in the BNC), it can also help to avoid including idiosyncratic usage by a single author.[26]

2.5 Window size

In the traditional, window-based approach to collocation candidate extraction, words within a certain window (or *span*) to either side of the word of interest are used for the calculation of frequencies and association measures. The size of this window is user-definable in any piece of software used for collocation candidate extraction and often defaults to 5, which is the received wisdom: "The ideal window size is different for each case. For the remainder of this paper, the window size, w, will be set to 5 words as a compromise; this setting is large enough to show some of the constraints between verbs and arguments, but not so large that it would wash out constraints that make use of strict adjacency" (Church/

[26] In fact, one can even use a fourth threshold on the isolated frequency of the collocate as offered by BNCweb. However, this threshold is only helpful to counter the low-frequency bias inherent in the Mutual Information association measure.

Hanks 1990, 24). Sinclair proposes the same distance: "The usual measure of proximity is a maximum of four words intervening" (Sinclair 1991, 170).

The window need not be symmetrical, so for the collocation types N + V and V + N (as used in OCD2), a restriction to the right and left context from the noun respectively may be a sensible choice.

Since we expect some sort of syntactic relationship to hold between the two partners in a collocation (at least in the base-oriented approach), it makes sense to stop looking at context across sentence boundaries. However, this presupposes either a pre-processing step with annotation of sentence boundaries or the use of (usually less elaborate) heuristics such as treating every instance of a full stop as a sentence boundary.

2.6 PoS patterns

In order to minimize the chance of including items that occur in the neighbourhood of a certain base but are not syntactically and semantically related, the use of PoS patterns has been proposed, most notably by Kilgarriff et al. (2004), who offer a commercial web-based tool (the *Sketch Engine*) that comes pre-programmed with a large range of such patterns that approximate syntactic relations (subject_of, object_of, etc.). The PoS patterns are written as regular expressions in the query language of the underlying corpus processing software, so for instance to identify premodifying adjectives of nouns, an adjective will only be considered to co-occur with the respective noun if both occur in the same pattern which may for instance specify that the adjective is optionally followed by further modifiers, which in turn are followed by the noun.[27]

For English, this approach seems to be quite successful and has been used in the creation of several dictionaries, in particular the ones published by Macmillan, including the MCD. It will miss out on items that are farther away than a few words intervening, but if the corpus is large enough, the quite effective filtering of noise will more than counter that effect.

For German with its less fixed word order and the larger distances between noun and verb in many types of clause, the PoS-pattern approach appears to be less convincing. A study on noun phrase case showed that there were serious issues:

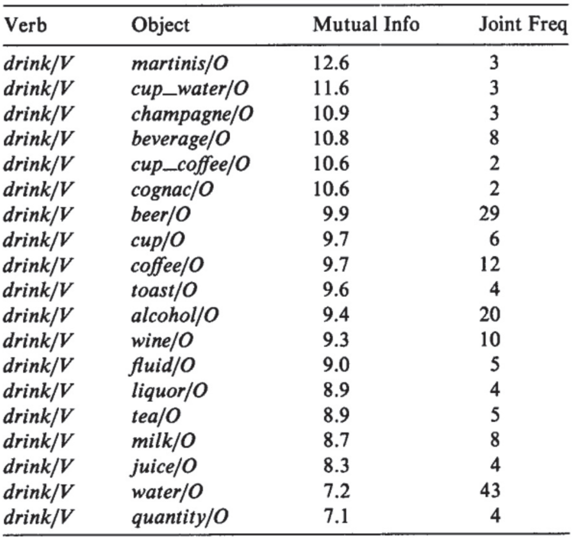
> The study has shown that the methods we have used are inferior to methods using richer linguistic inputs. This sets an agenda for us to improve German word sketches, by exploiting a lexicon to find noun gender, reviewing postagging and in particular, the tagset we have been using, and, in the longer term, using richer parsing strategies. (Ivanova et al. 2008, 2107)

In a very recent publication, Ambati/Reddy/Kilgarriff (2012, 2945) abandon the PoS-pattern approach for their analysis of Turkish in favour of a dependency-parsing approach similar to the one discussed below in the present paper: "Until now, word sketches have been generated using a purpose-built finite-state grammars [sic]. Here, we use an existing dependency parser."

[27] For the evaluation in section 4, we used the PoS patterns as available via the *Sketch Engine* in May 2012 for the Penn Treebank tagset. The only modification we made was that we grouped attributive and predicative adjectives together in order to obtain comparable results to our gold standard (see 4.1 for discussion).

3. Using dependency relations to improve the extraction of collocation candidates from corpora

The idea to use a parser in order to improve collocation extraction is by no means recent. Thus, for instance, Church/Hanks (1990) give the following list of objects for the verb *drink*, which was extracted from a parsed corpus:



Verb	Object	Mutual Info	Joint Freq
drink/V	*martinis/O*	12.6	3
drink/V	*cup_water/O*	11.6	3
drink/V	*champagne/O*	10.9	3
drink/V	*beverage/O*	10.8	8
drink/V	*cup_coffee/O*	10.6	2
drink/V	*cognac/O*	10.6	2
drink/V	*beer/O*	9.9	29
drink/V	*cup/O*	9.7	6
drink/V	*coffee/O*	9.7	12
drink/V	*toast/O*	9.6	4
drink/V	*alcohol/O*	9.4	20
drink/V	*wine/O*	9.3	10
drink/V	*fluid/O*	9.0	5
drink/V	*liquor/O*	8.9	4
drink/V	*tea/O*	8.9	5
drink/V	*milk/O*	8.7	8
drink/V	*juice/O*	8.3	4
drink/V	*water/O*	7.2	43
drink/V	*quantity/O*	7.1	4

Figure 3: Table from Church/Hanks (1990, 26) on objects of drink

However, until a few years ago parsing corpora of a substantial size made quite considerable demands on hardware and resulted in so many parsing errors that their use for the extraction of collocations would have been difficult to justify. With more accurate parsers[28] and more powerful hardware, the situation has however changed and the task becomes feasible. Seretan (2011) gives a very thorough account of a possible methodology with various evaluations and presents a tool for automated collocation candidate extraction from parallel corpora. While our own method differs in details (e.g. parser and grammar used[29]) and is limited to monolingual corpora, the general approach is very similar.[30]

[28] See Cer et al. (2010) for a discussion and figures on parsing accuracy. The data used in the present article was parsed with the Stanford Parser 1.6.9, originally described in Klein/Manning (2003).

[29] The parser used by Seretan does not always deliver full parses (Seretan 2011, 76).

[30] Lehmann/Schneider (2011) also use a very similar method for the extraction of different kinds of verb-attached prepositional phrases.

3.1 Description of methodology

The system for collocation candidate extraction presented here relies on dependency-annotated corpora. While it is in principle independent of language and parsing scheme, we shall use the Stanford Dependencies representation for English (de Marneffe/Manning 2008) in our discussion since it is the one used for the evaluation as well. One of the strengths of the Stanford Dependencies annotation scheme is that it is designed to have as many relations directly between content words as possible. For instance, let us consider the representation of sentence (1) in Figure 4:

(1) The girl was very beautiful. (CA3 1791)[31]

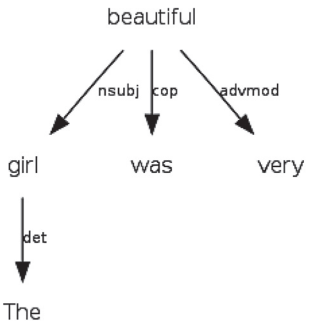


Figure 4: Graphical representation of sentence (1) in the Stanford Dependencies representation

We can observe a direct relation between *girl* and *beautiful* via an *nsubj* (nominal subject) relation while the copula is attached to *beautiful*, which is treated as the head of the sentence, via a *cop* (copula) relation. Thus in order to find predicative adjectives for nominal bases, we have to find an incoming nsubj relation from an adjective and can be sure – if the parser/tagger made no error – that we will include all predicative adjectives and only these in our extraction of collocation candidates.

To determine collocational strength, we apply different association measures to the following 2×2 contingency table:[32]

		collocate		
		coll	¬coll	
base	base	O_{11}	O_{12}	$= R_1$
	¬base	O_{21}	O_{22}	$= R_2$
		$= C_1$	$= C_2$	$= N$

[31] The letters and numbers in brackets indicate the position in the BNC (text and sentence ID).
[32] See Evert (2005; 2008) for a detailed explanation of such contingency tables.

For one specific base and one specific collocate, the table contains the co-occurrence frequencies of that specific base (*base*) and all other bases (¬*base*) with the specific collocate (*collocate*) and all other collocates (¬*collocate*) in one specific type of collocation, e.g. V + N. O_{11} is the number of co-occurrences of a specific base in the base slot with a specific collocate in the collocate slot in a specific type of collocation. R_1 is the overall frequency of that specific base in the base slot of that particular type of collocation. C_1 is the overall frequency of the specific collocate in the collocate slot of that type of collocation. N is the total frequency of that particular type of collocation.

3.2 Different combinations of dependency relations

In the interface presented in section 5 below, it is possible to combine several dependency relations to form one type of collocation. For instance, if we use the Stanford Dependencies model, we can combine the dobj (direct object) and the nsubjpass (nominal subject in passive clause) dependency relations into one type of collocation called V + N. Decisions as to which dependencies should be grouped together for best performance are of course language- and model-dependent. For the Stanford Dependencies for English, we can check which combinations of dependency relations best mirror the OCD2 gold standard (see discussion below) and use these as the default in our online tool *Treebank.info*.

Let us discuss one example here. If we are interested in noun-verb collocations and would like to make a difference between N + V and V + N as in OCD2, we would for instance have to decide whether we would want to include the xsubj (controlling subject) relation in the V + N collocation type or not. If we do, we will also find cases such as (2), where *care* is xsubj of *take*:

(2) Care has to be taken by a critic in any of these cases to describe works as definitely as possible. (A04 1536)

Most users would want to find such examples. However, we will also find cases such as the following, where *critic* is xsubj of *take*:

(3) The critic necessarily has to take a manifesto into account. (A04 1333)

Most users would prefer to find *critic* (if at all) in the candidate list for the collocation of the type N + V but not for V + N.

Thus, for best recall[33] we would have to include the xsubj relation in both the V + N and the N + V type of collocation, but this will cost us precision simply because the model does not make a distinction between active and passive xsubj in the same way as it does for nominal subjects (nsubj vs. nsubjpass) or clausal subject (csubj vs. csubjpass).

After testing various combinations of relations for the V + N collocation, it was decided that the best trade-off between precision and recall was achieved by using only the dobj and nsubjpass relations, but for any such trade-off, the decision is dependent on the exact use case.

[33] See the beginning of chapter 4 for a brief explanation of the terms *precision* and *recall*.

4. Evaluation

In order to obtain a profound assessment of the best settings of various parameters and of the accuracy of the various extraction methods, we shall use an automatic evaluation that compares the dataset extracted from the corpus against a so-called gold-standard dataset, i.e. a dataset that is considered to be a “perfect” solution to the task of collocation candidate extraction. Since we deemed it preferable to use a rather large dataset, we opted for OCD2 instead of a small sample annotated by lexicographers (see 4.1 below for details and issues associated with this method). The two measures we will be interested in for the performance evaluation are precision and recall, concepts borrowed from the domain of information retrieval (see for instance Russell/Norvig 2010, 869). Let us illustrate the two with an example:

The entry for *takeover* in OCD2 contains seven adjective[34] collocates: *attempted*, *proposed*, *hostile*, *company*, *corporate*, *communist*, *military*.[35] The top 10 collocation candidates for a given set of parameters (co-frequency threshold 2, lemmatized, dependency-based, t-score) extracted from the corpus contains the following adjectives: *hostile*, *communist*, *reverse*, *military*, *propose*, *successful*, *contest*, *imminent*, *chinese*[36], *foreign*.

Of the 10 candidates, 3 are actual collocates in the gold standard document, so our measure of precision at a list length of 10 is 3/10 or 0.3. Since there are 7 true collocates in the dictionary, our measure of recall at a list length of 10 is 3/7 or roughly 0.43. Thus precision gives us the proportion of the “good” collocations within our list of collocation candidates while recall gives us the proportion of all “good” collocations from the gold standard that were actually found by the system.

Using a large gold-standard dataset allows us to verify both precision and recall automatically for each entry in the dictionary, while studies with manual evaluation often focus mainly on precision (e.g. Seretan 2011, 125) and use N-best lists over the whole corpus as the basis of evaluation, i.e. they look at the most strongly associated words in the whole corpus and not for a set of bases (e.g. Evert/Krenn 2005, Seretan 2011).

4.1 OCD2 as gold standard

As mentioned above, we shall use the CD-ROM version of OCD2 as the gold standard in our evaluation of the collocation candidate extraction, i.e. we shall assume, for the purpose of this experiment, that due to the extensive manual work performed by the lexicographers during the creation of the dictionary, it contains perfect lists of collocates for a given base in the sense of exactly the collocates any lexicographer would want to include in a dictionary entry. Such a simplification is necessary in order to compare the different approaches against a large dataset, even though one has to be aware that OCD2 is not necessarily

[34] See the brief discussion in 4.1 on the problem of nouns turning up in the adjective slot of ADJ + N collocations in OCD2.

[35] The fact that not all of them may be classified as collocations by all researchers (for instance, *company takeover* could be regarded as a compound instead) is of no particular relevance to the discussion here.

[36] The lemmatization process automatically converts all words to lower case.

authoritative. Thus Herbst/Klotz (2009, 241f) show that the overlap between a small sample taken from three collocation dictionaries, one of which is OCD1, is smaller than one may expect.[37]

There are a few more issues in the comparison that have to be borne in mind. First of all, the *Oxford English Corpus*, which was used for the compilation of the dictionary, is much larger (2 billion words) than the *British National Corpus* (BNC), which we used for the collocation candidate extraction. Secondly, while our corpus contains only British English mainly of the 1980s and 1990s, the *Oxford English Corpus* contains texts "from up-to-date sources from around the world" (OCD2, vi). Thus it is impossible to find some of the collocations found in OCD2 simply because they do not occur in the corpus used for the present study.

Furthermore, the dictionary sports many multi-word collocates, and while it might have been relatively easy to identify items such as phrasal verbs with the help of the dependency-annotated corpus, it would of course have skewed the results compared to the window-based and the PoS-pattern approaches, so multi-word collocates are not used in the evaluation.

A further complication is introduced by the fact that in the dictionary, the slash (/) character can separate spelling variants (*grey/gray*), words and their abbreviations (*television/TV*) or be part of one collocate (*20/20* in *20/20 vision*[38]). Since it is difficult to decide automatically to which type any of the items with internal slash belong, all collocates with slashes were ignored in the evaluation.

In addition, the dictionary often presents semantically similar collocates in a list terminated by "etc.", where the collocation candidate extraction may find further items that would belong in this list but simply do not figure in the dictionary. In the entry for *accent*, for instance, OCD2 lists *American, British, English, French, etc.* while our collocation candidate extraction in the BNC also identifies *Irish*, *Scottish*, *German* and *Australian* as strongly associated with *accent*, which is of course correct but will reduce precision in the evaluation simply because the dictionary compilers decided not to include them.

Finally, OCD2 is very generous as to what it counts as an adjective in ADJ + N collocations in that the compilers "quite cheerfully put into the ADJ. slot items that are actually pre-modifying nouns, such as *tax benefit* and *takeover bid*" (Lea 2007, 269 on OCD1). This grammatically lax methodology will again result in lower precision and recall values since the automatic systems only extract collocates tagged as adjectives for ADJ + N collocations.

All issues we have discussed so far can explain why precision (and recall at N to some extent) is relatively low in the comparisons carried out below. We expect all approaches to suffer from these problems to the same extent, though, so that the comparison is still valid.

[37] One reason may be the slightly different concept of collocation used in these dictionaries.

[38] The example of *20/20 vision* serves as a very good example to warn readers of the dangers of relying purely on automatic procedures. Hardly any system would have picked up *20/20* given that it can hardly be analysed as an adjectival or nominal modifier of *vision*. The extensive work with concordances described in Lea (2007) is probably what is responsible for the fact that the collocation was included.

However, there is one issue that may actually skew the comparison of the approaches if we use OCD2 as gold standard. Given that OCD1/2 were created with either the window-based or the PoS-pattern approach to extract collocation candidates, there may be a bias against the dependency-based approach. The reason is that we would expect lexicographers to be more prone to use top-of-the-list material in their entries than material found further to the bottom of a list or not at all in the list when they make use of summary pages such as word sketches or collocation profiles, as was done for high-frequency items in the creation of OCD1 (see Lea 2007, 264),[39] even though we expect this effect to be somewhat smoothed out by the manual intervention of the lexicographer.

Keeping in mind the problems mentioned above, it is nonetheless sensible to rely on OCD2 as a gold-standard dataset due to its high quality and its sheer size. Even with the multi-word units and items with slashes ignored, the number of collocations found in the dictionary is rather impressive and allows for an evaluation at an extremely large scale. To keep the number manageable, we will include in our evaluation only three types of collocation, ADJ + N (both attributive and predicative use of the adjective), V + N (corresponds roughly to verb plus direct object) and N + V (corresponds roughly to subject plus verb); as mentioned in 1.2, in all these, the noun is treated as the base by the dictionary, so it is sufficient to use the noun entries. Of the 5,306 noun entries in the dictionary that contain at least one of these types, 5,058 contain a total of 86,565 adjective collocates, 4,481 contain a total of 36,670 verb collocates in the V + N type, and 1,383 contain a total of 4,797 verb collocates in the N + V type. So all in all, our gold standard contains 128,032 collocations and thus enables us to make sound comparisons of the various parameters and extraction methods not only for precision but also for recall. Having at our disposition such a large number of gold standard collocations allows us to measure precision and recall for each individual base, which mirrors the work of a lexicographer. The numbers presented below are averages over all bases in the respective testbed.

4.2 Window size

In order not to compare the dependency method to a deliberately bad baseline, it was planned to use the “best” window size for each collocation type tested, even though many users rely on a 5-word window as quoted above and often do not change the window size depending on which type of collocation they look for. Thus a database with collocations in window sizes of 1 to 5 (i.e. 0 to 4 items intervening) was created and the resulting lists were matched against the lists from the OCD. For ADJ + N, the window was symmetrical, for V + N and N + V, only the right-hand or left-hand context was taken into account. The association measure chosen was t-score, the frequency threshold 5.

[39] Kilgarriff et al. (2010, 373) state that “[t]he OCD was compiled by lexicographers studying corpus evidence but without using word sketches”, so we can be sure that OCD2 made no use of the *Sketch Engine*. Nonetheless, the figure given by Lea (2007, 264) of the view generated by the collocation extraction system used for the creation of OCD1 seems to suggest that a combination of both methods may have been used since the system makes a distinction between left-hand side and right-hand side for some items (e.g. verb collocates for nouns), but not for others (e.g. adjectives), but it cannot be ruled out that simply both symmetrical and asymmetrical windows were used.

Let us start by looking at noun-adjective collocations:

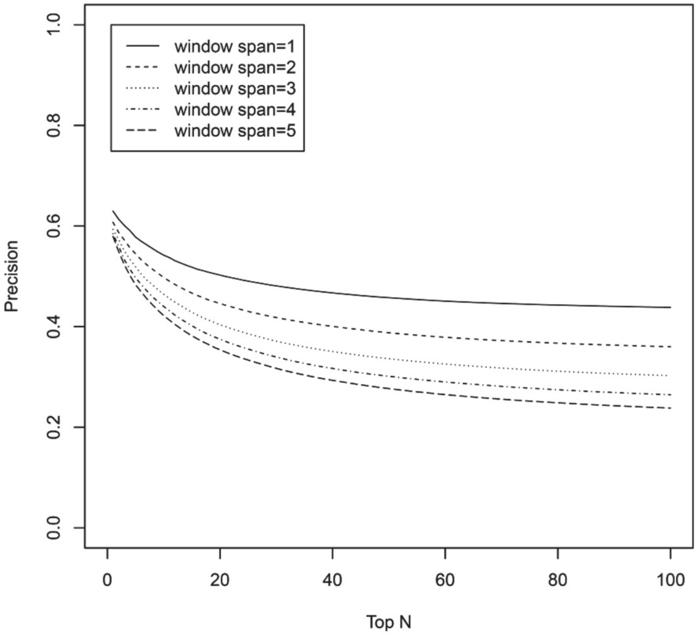


Figure 5: ADJ + N: Precision with t-score; frequency threshold = 5

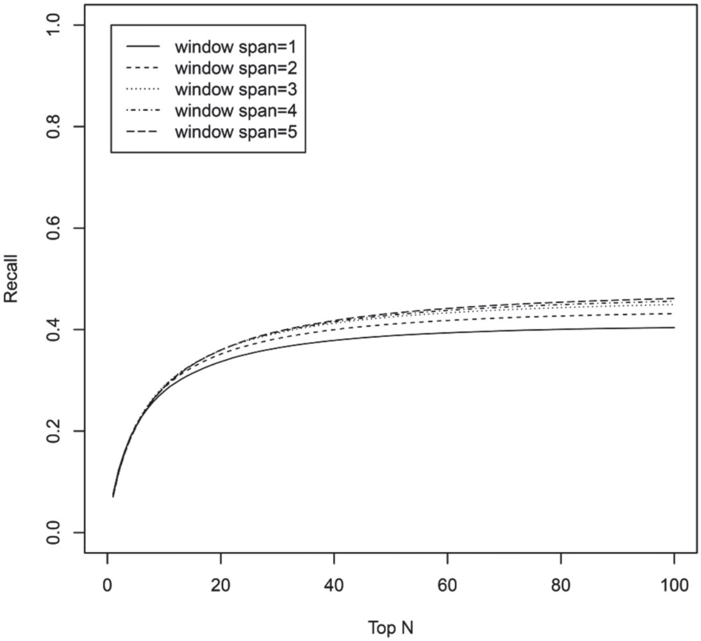


Figure 6: ADJ + N: Recall with t-score; frequency threshold = 5

Not surprisingly, precision improves with smaller window sizes, i.e. unwanted material does not get into the database. On the other hand, some relevant items are not found (or not found sufficiently frequently to reach the frequency threshold) with a windows size of one, so recall improves with a bigger window size. However, given that recall does not differ dramatically between the different approaches, it might be reasonable to use smaller windows in order to benefit from the higher precision in some lexicographic applications.

For V + N collocations (i.e. direct objects in active clauses and subjects in passive clauses in OCD) the situation looks different. At first sight, the precision graph seems to imply that if we look at collocation candidate lists longer than 35, window size 1 delivers best results, whereas it is not as good as window size 2 with shorter lists. However, this interpretation is to some extent an artefact of the way our data is plotted since for window size 1, many lists are shorter than 35 items and the value of the longest available list is then used for the remainder of the graph. We can in fact observe that the high precision comes at a price when we look at the recall graph, where the recall of window size 1 is drastically lower than that of all other window sizes. From a linguistic point of view the low score of window size 1 is of course not very surprising, given that most English nouns need a determiner in order to form a grammatical noun phrase.

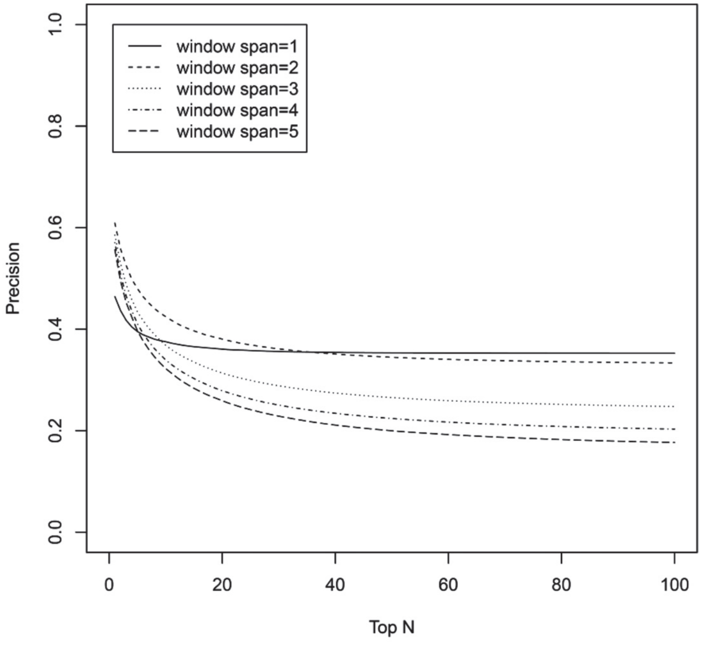


Figure 7: V + N: Precision with t-score; frequency threshold = 5

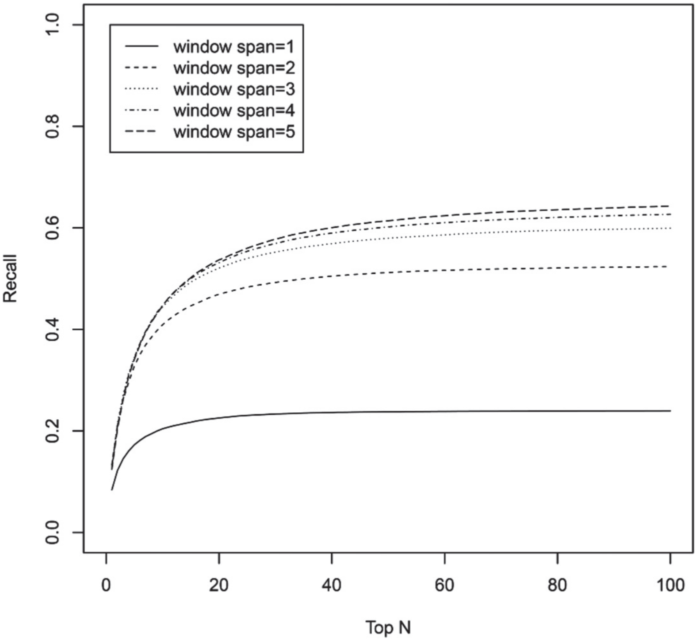


Figure 8: V + N: Recall with t-score; frequency threshold = 5

For the type N + V (i.e. active clause subjects and *by*-phrases in passive clauses in OCD), the results are similar to the ones for V + N, only with smaller differences between the various window sizes. Again, the low recall of smaller windows is related to grammatical features of English, where auxiliary verbs or adverbs often occur between the subject and the main verb.

In sum, we can observe that the choice of window size is a trade-off between precision and recall, so instead of selecting only one for comparison, we shall use a small and a large window to compare to the dependency-based and the PoS-pattern approaches. For ADJ + N, the window sizes shall be 1 and 5, for V + N and N + V we shall use 2 and 5 as lower and upper bound for the reasons just outlined above.[40]

4.3 Co-frequency threshold

A co-frequency threshold can be applied to the candidate lists of all three approaches. Here we take a look at adjective-noun collocations extracted by the dependency-based approach, but results are very similar for the other approaches and types of collocations. The associ-

[40] Klotz (2000, 76–84) shows a similar effect for actual instances of collocations and collocation candidates. He manually compares the results of a small and a large window for eight V + N collocations and shows that while he retrieves a larger number of relevant results with a larger window (i.e. better recall) the number of syntactically unrelated candidates increases at the same time (i.e. worse precision).

ation measure chosen was log-likelihood. We chose log-likelihood over t-score for this particular test because of the "built-in" frequency threshold of t-score mentioned above (see section 2.4).

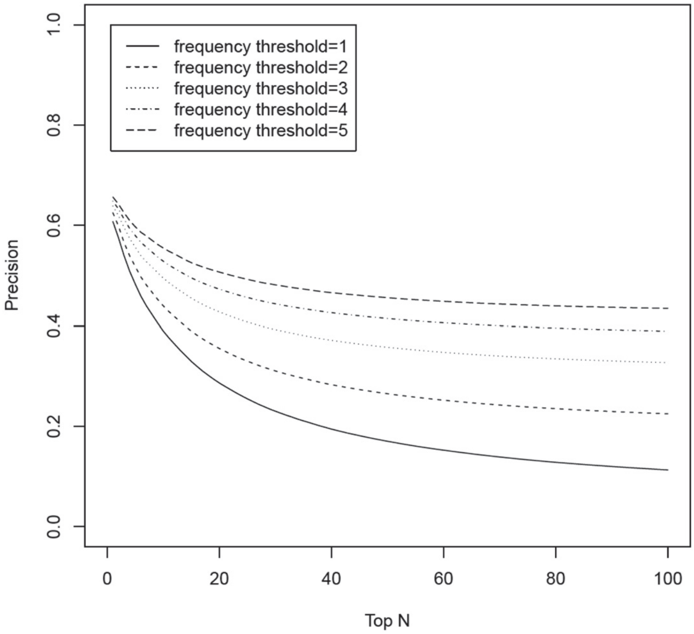


Figure 9: ADJ + N: Precision with log-likelihood; dependency-based approach

As was to be expected, higher co-frequency thresholds lead to higher precision values, as rare and untypical collocation candidates are filtered out. However, we have to bear in mind that part of this effect might be an artefact from the creation process of OCD2 which may have included the application of a co-frequency threshold. As with window size, higher precision comes at the cost of lower recall.

The more candidates we filter out by applying a higher co-frequency threshold, the higher the risk of dismissing true collocates. What is interesting, however, is the behaviour for a co-frequency threshold of 1. For N-best lists with N ≤ 50, a threshold of 1 counter-intuitively results in a lower recall than a threshold of 2. And for N-best lists with N ≤ 19, a threshold of 1 performs even worse than a threshold of 3.

This counter-intuitive behaviour can be explained by strongly associated low-frequency items. Let us illustrate this by looking at the 20-best lists of the noun *acceleration* for co-frequency thresholds of 1 to 3.

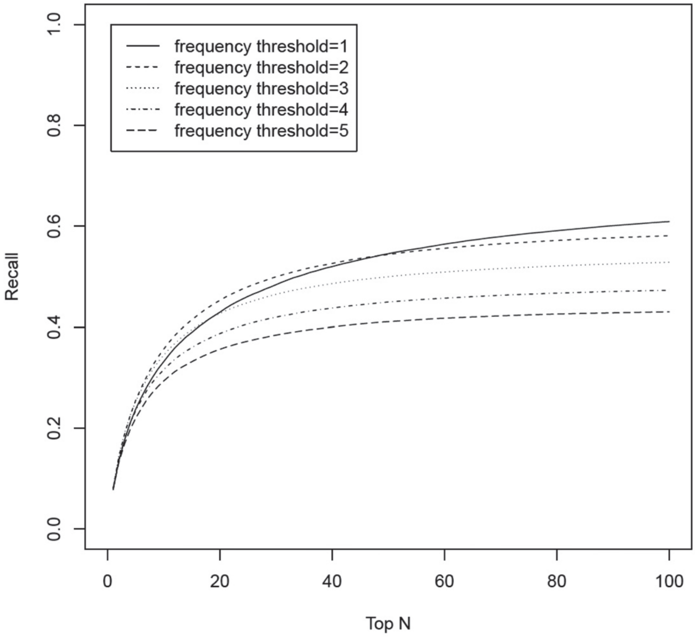


Figure 10: ADJ + N: Recall with log-likelihood; dependency-based approach

Rank/cofreq	**1**	**2**	**3**
1	gravitational	gravitational	gravitational
2	**rapid**	**rapid**	**rapid**
3	60mph	60mph	60mph
4	tidal	tidal	tidal
5	constant	constant	constant
6	mass	mass	mass
7	linear	linear	linear
8	brisk	brisk	brisk
9	relative	relative	relative
10	horizontal	horizontal	horizontal
11	aortic	aortic	tremendous
12	tremendous	tremendous	maximum
13	inertial	inertial	**fast**
14	centrifugal	centrifugal	vertical
15	rip-snorting	maximum	**poor**
16	steadyish	**fast**	**sudden**
17	0–60	permissible	**slow**
18	maximum	vertical	substantial
19	**fast**	longitudinal	possible
20	permissible	**poor**	less
Recall at 20	**28.6%**	**42.9%**	**71.4%**

Table 1: Top 20 collocations with co-frequency thresholds ranking from 1 to 3

In the table, collocates from the OCD are set in bold, candidates with a co-frequency exactly corresponding to the threshold (i.e. being filtered out in the next column) have a grey background. As can be seen, the 20-best list for a co-frequency threshold of 1 contains three items with a co-frequency of 1. These three items have been filtered out in the 20-best list for a co-frequency threshold of 2, allowing *poor* to enter the list and improve both precision and recall. For a co-frequency threshold of 3, five further items have been filtered out, introducing two more gold collocates, *sudden* and *slow*, and further improving precision and recall. However, these two collocates both have a co-frequency of 3 and are therefore not included in the 20-best list for a co-frequency threshold of 4, resulting in lower precision and recall values.

As with window size, we can observe that the choice of co-frequency threshold is a trade-off between precision and recall. We cannot give a hard and fast rule as to its best setting: Corpus size, association measure, and the intended lexicographic use can all influence a sensible choice of this parameter.

4.4 Association measures

As mentioned above, there is a whole range of association measures available to lexicographers and the choice is, to some extent, a matter of personal preference. Evert/Krenn (2005) present a methodology for the evaluation of association measures for specific purposes that relies on manual annotation of a sample of collocation candidates as a reference point. Here, the approach shall be to compare the lists of collocation candidates created with various measures to the OCD as our gold standard. It has to be stressed, though, that this methodology is highly likely to pick out not necessarily the “best” association measure for lexicographic work but rather the association measure used in the creation of the dictionary.

The following graphs were generated using the dependency-based collocation candidate extraction for V + N collocations, but the picture is similar for other collocation types and for other methods of extraction. All graphs feature a line called Oracle that represents the best possible ordering of the extracted list, i.e. the ideal association measure to represent our gold standard.

Let us first take a look at the association measures used without a frequency threshold (Fig. 11 and 12).

We can observe that t-score outperforms all other measures at a list length of 20, interestingly closely followed by co-frequency. The reason for this advantage of t-score may be the built-in frequency filter mentioned above. With longer lists, MI3 and co-frequency overtake t-score. Dice turns out to be relatively low in both precision and recall, particularly with short lists. Its strong low-frequency bias makes Mutual Information (MI) the least helpful association measure for a use without frequency threshold.

If we set the co-frequency threshold at 5, as in Fig. 13 and 14, the picture changes dramatically. All association measures show a similar distribution and are thus almost indistinguishable in the graph, even with short lists (the average list is only about 31 items long for this type of collocation). At a list length of 20, there is no significant difference between MI3 and t-score, with MI3 slightly in the lead. However, the differences in both precision and recall are relatively small at short list lengths and at best marginal at list lengths of

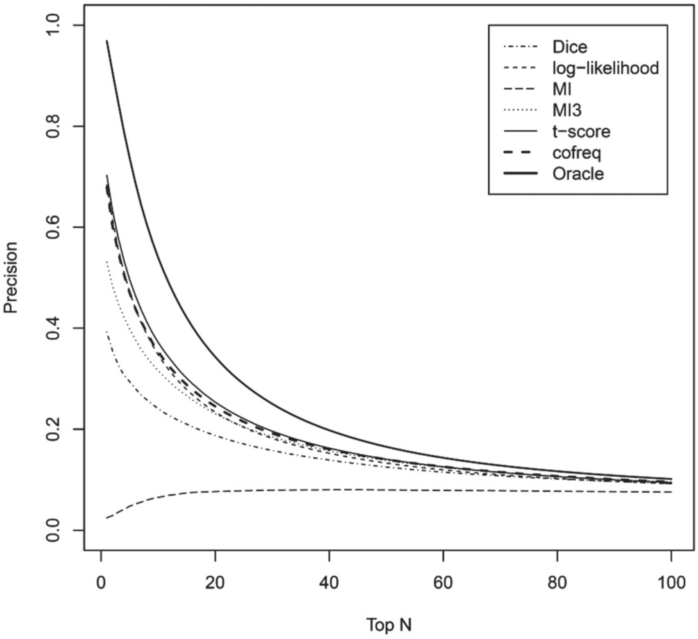


Figure 11: V + N: Precision without frequency threshold; dependency-based approach

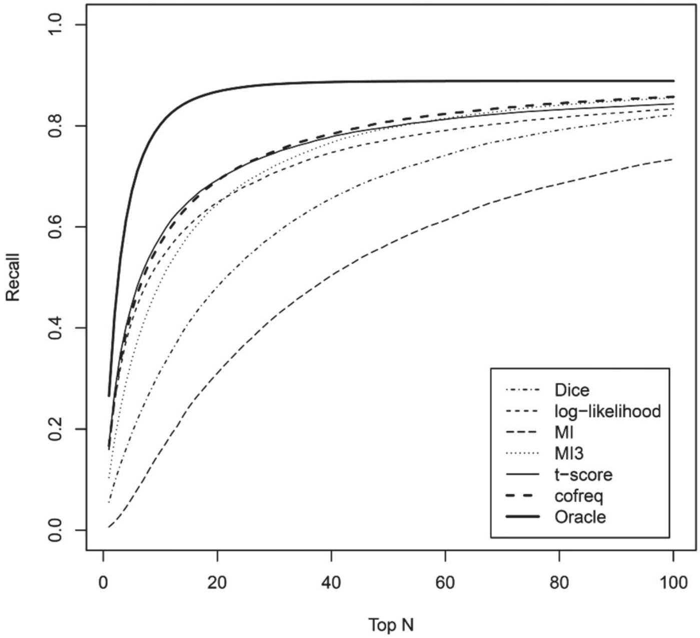


Figure 12: V + N: Recall without frequency threshold; dependency-based approach

about 50 for all the association measures tested, so a frequency threshold will successfully counter even the strong low-frequency bias of MI for longer lists.[41]

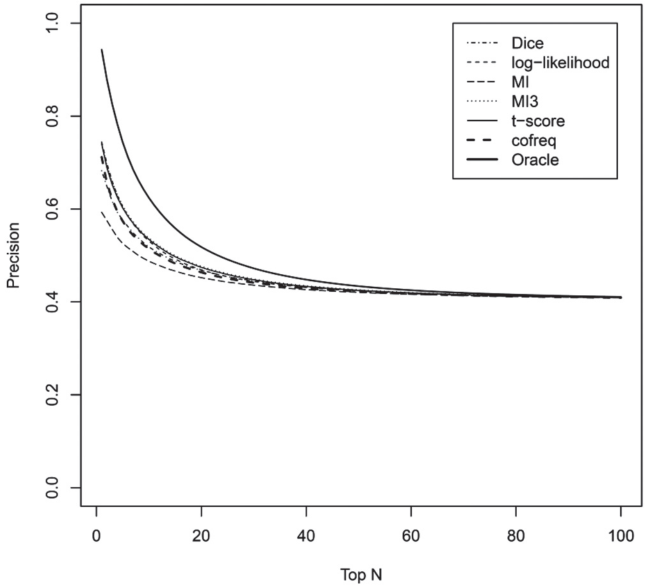


Figure 13: V + N: Precision with frequency threshold = 5; dependency-based approach

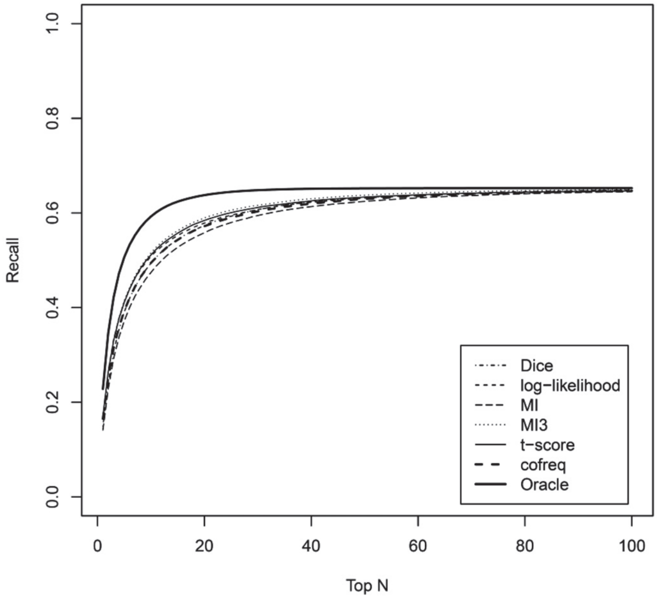


Figure 14: V + N: Recall with frequency threshold = 5; dependency-based approach

[41] At a candidate list length of 10, MI is still significantly lower than the other measures in both precision and recall.

We can thus conclude that – at least for the majority of the collocations in our gold standard – the differences in association measures are negligible for longer lists of collocation candidates as soon as a frequency threshold is applied. Without a frequency threshold, the "built-in" frequency threshold of t-score makes it the most accurate association measure for our purposes, and co-frequency is surprisingly accurate, too, which seems to indicate that high-frequency collocates are more numerous in the gold-standard than low-frequency ones.

4.5 Dependency vs. other approaches

4.5.1 Length of collocation candidate lists

One effect of choosing a specific approach to the identification of collocation candidates that is of immediate relevance to lexicographic work is the impact on the amount of data the lexicographer has to sift through. Let us demonstrate this by comparing the average length of candidate lists for collocations of the type V + N using different approaches and co-frequency thresholds.

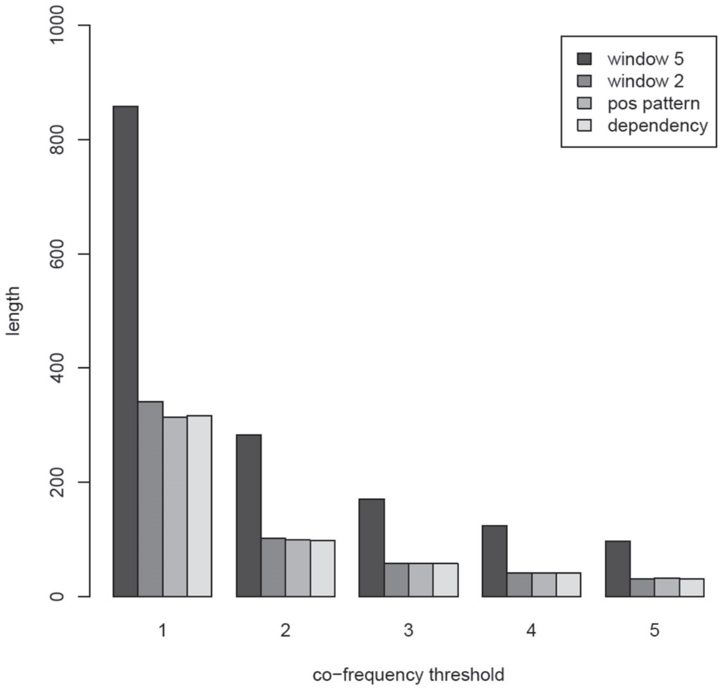


Figure 15: V + N: average length of collocation candidate lists

The first and most obvious observation we can make is that higher co-frequency thresholds lead to shorter candidate lists as more and more items are filtered out. As was to be expected, smaller window sizes also lead to shorter candidate lists as fewer potential candidates are taken into account. Probably the most interesting observation to be made is that there is scarcely any difference in average length of collocation candidate lists between the dependency-based approach, the PoS-pattern approach and the window-based approach with a window size of 2. That means lexicographers can expect roughly the same number of candidates for any of those three approaches. That also means that any differences in precision or recall between those collocation candidate lists have to be a consequence of the different extraction methods applied and are not due to simply longer or shorter lists.

4.5.2 Precision and recall

As mentioned at the beginning of chapter 4, the two most interesting values to compare are precision and recall for N-best lists of collocation candidates. Let us first take a look at ADJ + N collocations.

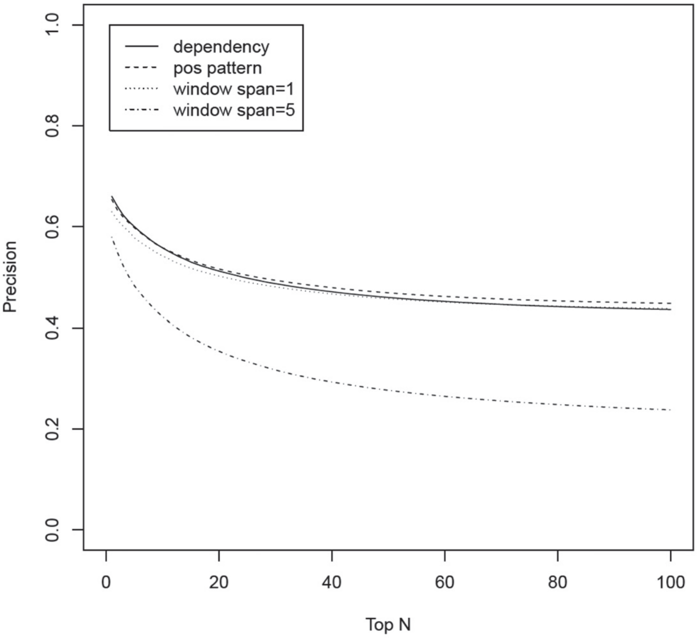


Figure 16: ADJ + N: Precision with t-score; frequency threshold = 5

For precision, it does not really seem to matter whether one chooses a window-based approach with a window span of 1, a PoS-pattern-based approach or a dependency-based approach. Although the PoS-pattern-based approach is slightly in the lead for longer N-best lists, the difference to the dependency-based approach is not significant at N = 50 and verges on the border of significance (p = 0.0145) at N = 100.[42] Using a window-based approach with the popular window size of 5, however, leads to significantly lower precision values from the beginning (p < 2.2e-16 both at N = 20 and at N = 100).

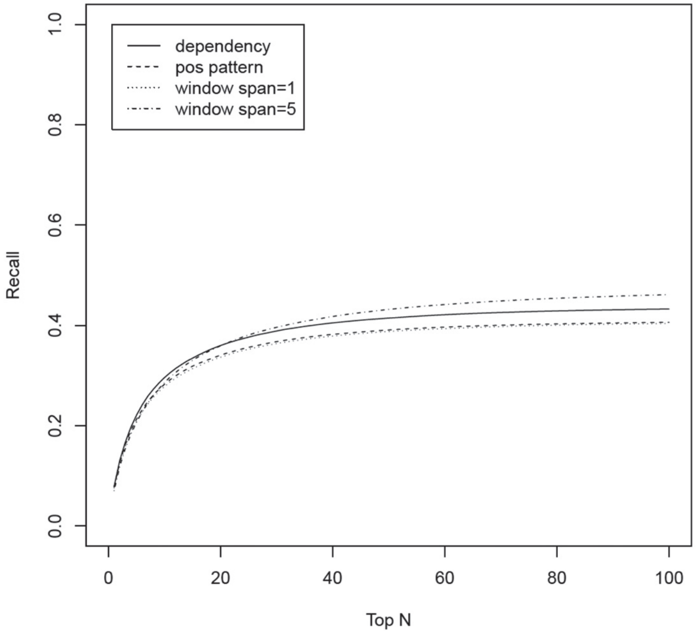


Figure 17: ADJ + N: Recall with t-score; frequency threshold = 5

Looking at the recall values gives us a different picture. For N-best lists with N ≤ 20 the dependency-based approach yields the highest recall values (at N = 20 significantly higher than PoS pattern and window size 2), for larger Ns it is overtaken by the window-based approach with window size 5 (not significant at N = 30, but at N = 50 with p = 0.00025). This is probably due to the much longer candidate lists resulting from that approach. The PoS-pattern-based approach and the window-based approach with window size 1 have almost identical recall values.[43]

[42] All p-values in the present chapter were calculated using the Asymptotic Wilcoxon-Mann-Whitney Rank Sum Test.

[43] This is in line with Heid et al.'s observation, who compared a PoS-pattern approach to a full syntactic parse for the extraction of German juridical phraseology from corpora: "In a preliminary experiment, we compared the collocation candidate lists from both approaches. Contrary to a widespread assumption, it is not as much precision, but rather recall which is enhanced through the use of parsed data." (Heid et al. 2008, 138)

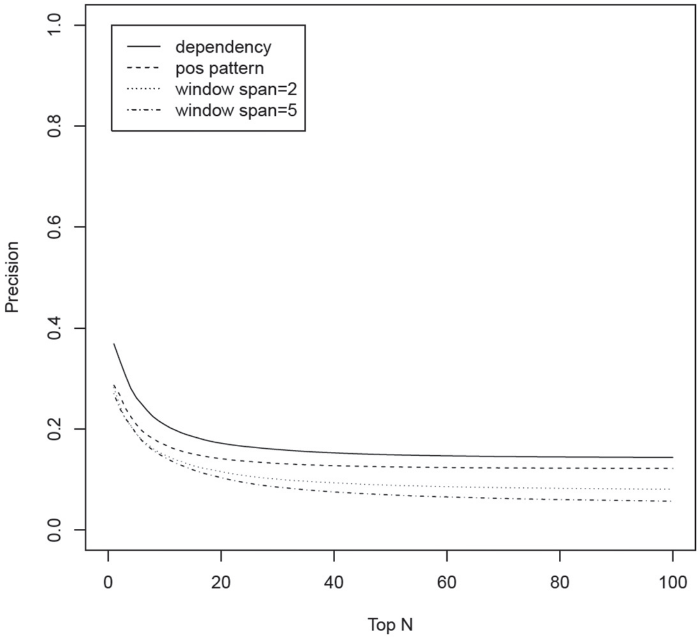


Figure 18: N + V: Precision with t-score; frequency threshold = 5

Precision values for N + V collocations are much lower than for ADJ + N.[44] We can easily recognize a ranking of the different approaches, with the dependency-based approach significantly in the lead, followed by the PoS-pattern approach, which in turn is significantly better than the window-based approach with window size 2 and with window size 5.

Regarding recall, the window-based approach with window size 5 leads to significantly better recall values for N = 50 and N = 100 than all other approaches. For shorter lists, the dependency-based approach performs best, for longer lists second-best. Compared to the window-based approach with window size 2, the dependency-based approach is significantly better at N = 20 but not at N = 100. The PoS-pattern-based approach yields the worst recall values, performing significantly worse than the dependency-based approach (Fig. 19).

The precision ranking of the different approaches for V + N collocations is the same as for N + V collocations, albeit on a much higher level, so the dependency-based approach is significantly better than all other approaches, both at N = 20 and at N = 100 (Fig. 20).

[44] A cursory glance at the results seems to suggest that the reason is the lower frequency of many of the collocations in the dictionary and the often short lists of collocates (many entries only contain one collocate of that type).

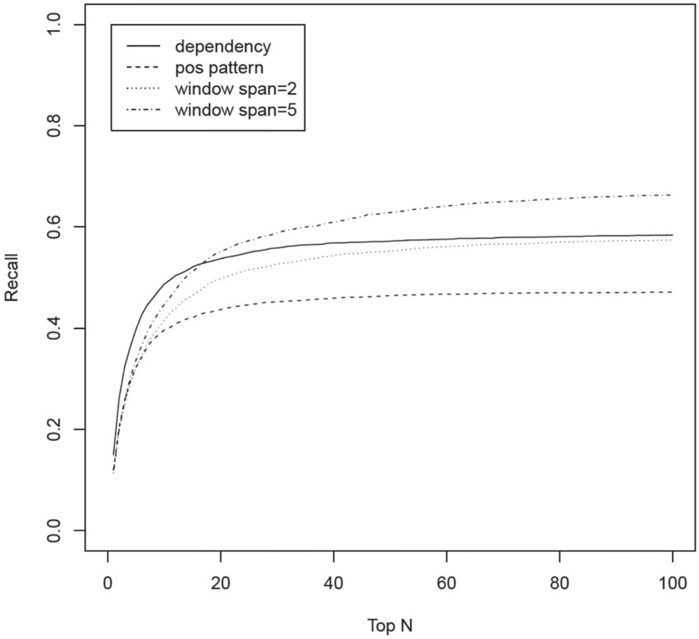


Figure 19: N + V: Recall with t-score; frequency threshold = 5

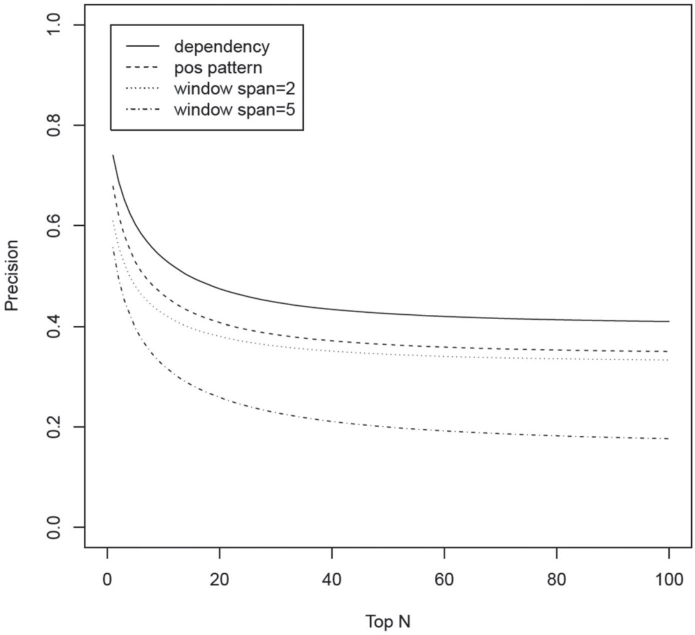


Figure 20: V + N: Precision with t-score; frequency threshold = 5

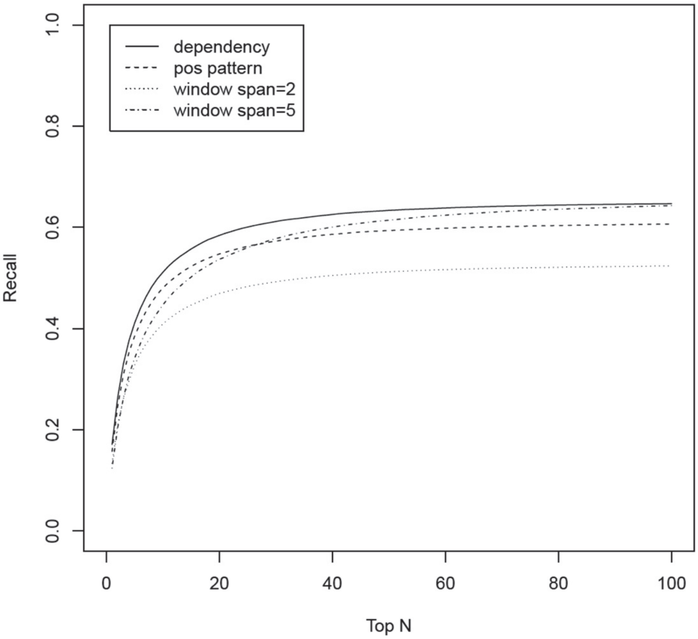


Figure 21: V + N: Recall with t-score; frequency threshold = 5

For recall, the dependency-based approach is significantly better than the PoS-pattern approach both at N = 20 and at N = 100. It also beats the window-based approach with window size 5; at N = 20, and N = 30 the difference is significant, at N = 50, the difference verges on the border of significance (p = 0.0234), and at N = 100, the difference is not significant.

In sum, we can see that the dependency-based collocation candidate extraction performs best for most tasks if users want to achieve a balance between precision and recall. For some tasks, other approaches may deliver better results for either precision or recall, but usually at a high cost to the performance in the other measure.

5. Practical implementation

In order to apply the method of collocation candidate extraction from a dependency-parsed corpus, users can make use of the *Treebank.info* project freely available online at <http://treebank.info>. The present chapter will give a brief overview of this web-application, the steps necessary to prepare a collocation database and the results users can expect.

5.1 *Treebank.info* as technical basis

The original idea of the *Treebank.info* project is to make available parsed corpora to less computationally-minded linguists. The application interacts with the user via a web-interface where the user can upload his/her own corpora and have them processed and then query them via a relatively easy-to-use graphical interface. A few publicly available corpora are pre-loaded as well. The beauty of using *Treebank.info* as the technical basis for the collocation candidate extraction task presented here lies in the fact that the upload mechanism and the processing pipeline are already available, so that users can benefit from the high-performance, highly-scalable back-end that can parse large corpora in short periods of time without having to worry about the technical details.[45]

5.2 User-defined collocation databases

As mentioned in 3.2, users can group different syntactic relations together to form one type of collocation and can combine various collocations in their own collocation database. *Treebank.info* comes with a set of default settings (the ones used for the present paper and a few more) for each annotation scheme and all publicly available corpora have been processed with this scheme, so even users who do not wish to go into the details of the annotation scheme (in our case the Stanford Dependencies mentioned above) can profit from the improvement the dependency-based collocation candidate extraction offers over the other methods. If the user wants to, however, he/she can edit not only the dependency relations used (and their direction) but also apply Part-of-Speech-based filters. All databases are created both for the lemmatized and the unlemmatized variants of each word. Figure 22 is a screenshot from *Treebank.info* that shows the summary of the collocation database about to be created (Fig. 22).

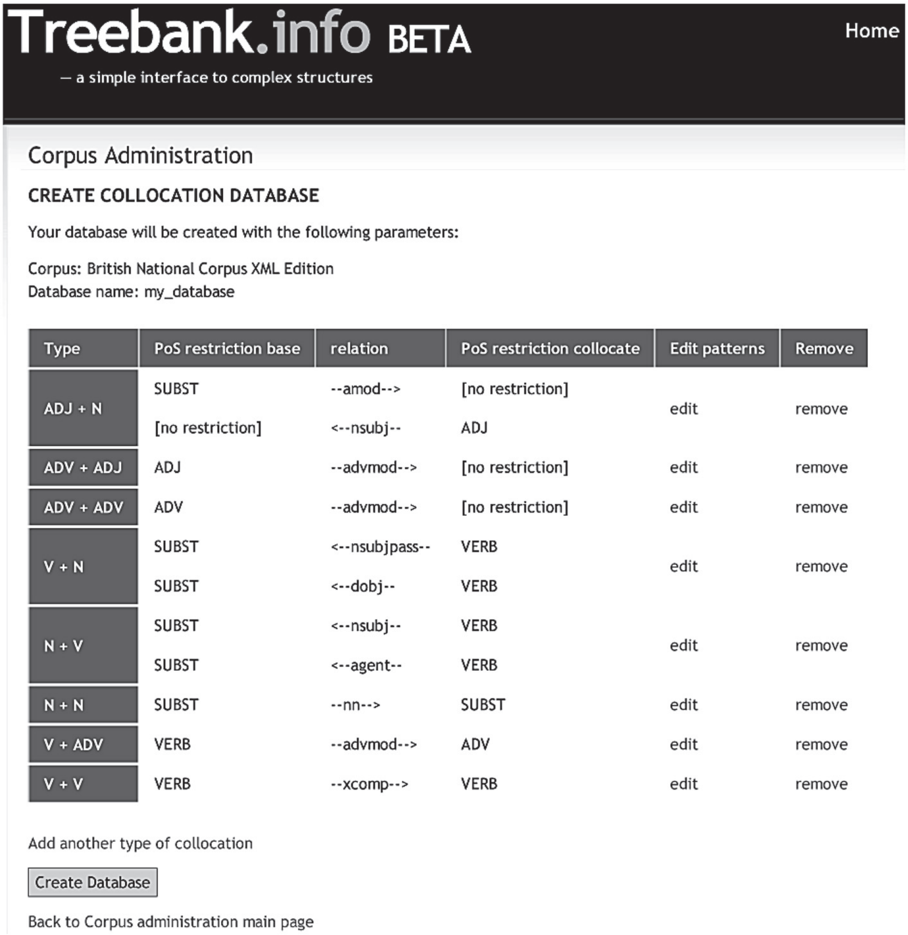
5.3 Presentation of results

For the presentation of the results, a view similar to the word sketches offered by the *Sketch Engine* was chosen, where a configurable number of results will be displayed for each type of collocation. As of summer 2012, the user can choose between lemma and word form, select a part of speech, choose between various association measures and set a co-frequency threshold.

The screenshot in Figure 23 shows 10 collocation candidates of each type with the noun *time* as the base or collocate. Given the arguments brought forward by Rundell (2011) and applied in MCD, we would find it unsatisfactory to only include candidates for which the queried item acts as a base in Hausmann's (2004) sense, since it appears to be less straightforward to decide a priori which element should be regarded as base and which as collocate simply on the basis of word class (see the brief discussion in 1.2 above).[46] Due to the fact

[45] Technical details can be found in Proisl/Uhrig (2012).

[46] In the future, it is planned to allow for a configurable view, i.e. to let the user decide which types



Corpus Administration

CREATE COLLOCATION DATABASE

Your database will be created with the following parameters:

Corpus: British National Corpus XML Edition
Database name: my_database

Type	PoS restriction base	relation	PoS restriction collocate	Edit patterns	Remove
ADJ + N	SUBST [no restriction]	--amod--> <--nsubj--	[no restriction] ADJ	edit	remove
ADV + ADJ	ADJ	--advmod-->	[no restriction]	edit	remove
ADV + ADV	ADV	--advmod-->	[no restriction]	edit	remove
V + N	SUBST SUBST	<--nsubjpass-- <--dobj--	VERB VERB	edit	remove
N + V	SUBST SUBST	<--nsubj-- <--agent--	VERB VERB	edit	remove
N + N	SUBST	--nn-->	SUBST	edit	remove
V + ADV	VERB	--advmod-->	ADV	edit	remove
V + V	VERB	--xcomp-->	VERB	edit	remove

Add another type of collocation

Create Database

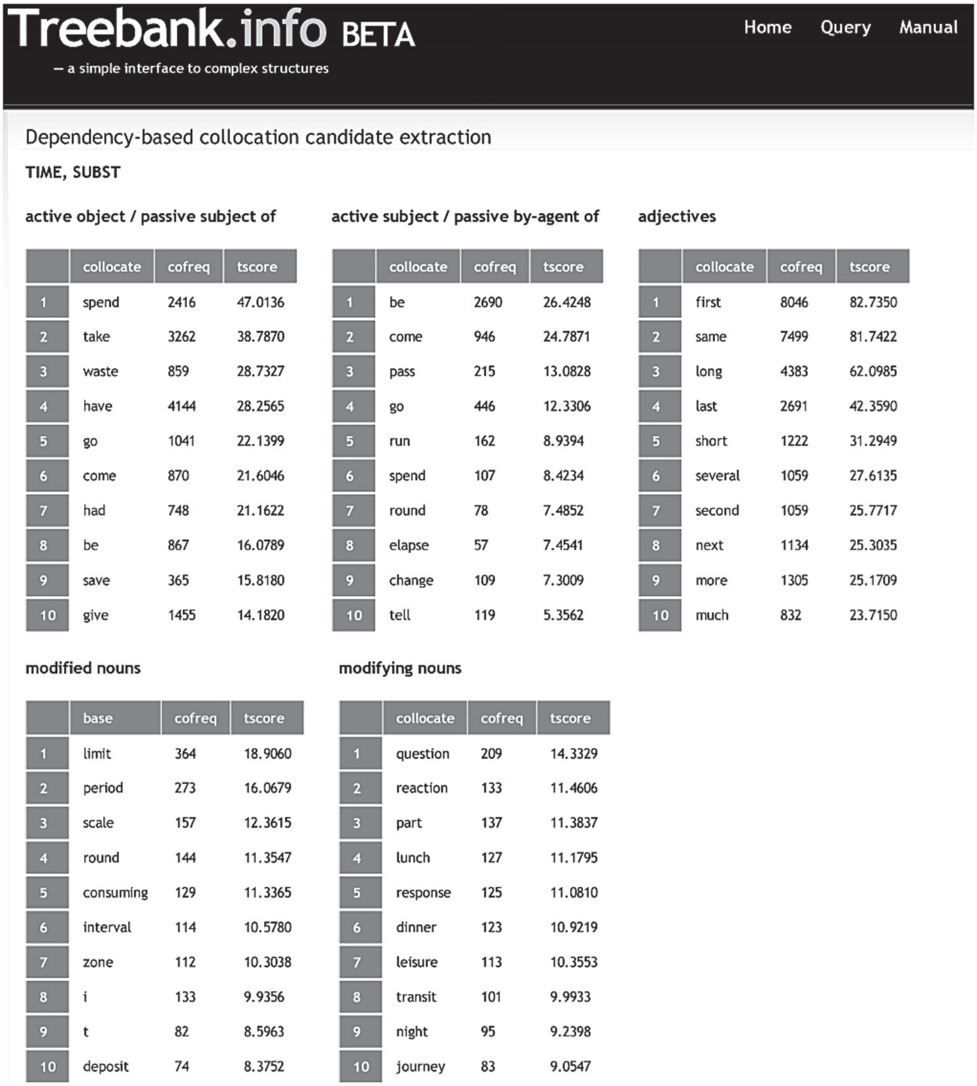
Back to Corpus administration main page

Figure 22: Collocation database settings in *Treebank.info*

that all co-occurrences are pre-calculated, the response time of the system is much shorter than for systems that have to calculate collocation databases for individual words on the fly, such as *BNCweb*, which in turn is much faster than concordancers such as *AntConc* due to heavy use of indexing.[47]

of collocation should be listed for a query with a given PoS. The *Sketch Engine* already offers such options.

[47] This results of course in increased storage demands on our server compared to such other solutions.



Dependency-based collocation candidate extraction

TIME, SUBST

active object / passive subject of

	collocate	cofreq	tscore
1	spend	2416	47.0136
2	take	3262	38.7870
3	waste	859	28.7327
4	have	4144	28.2565
5	go	1041	22.1399
6	come	870	21.6046
7	had	748	21.1622
8	be	867	16.0789
9	save	365	15.8180
10	give	1455	14.1820

active subject / passive by-agent of

	collocate	cofreq	tscore
1	be	2690	26.4248
2	come	946	24.7871
3	pass	215	13.0828
4	go	446	12.3306
5	run	162	8.9394
6	spend	107	8.4234
7	round	78	7.4852
8	elapse	57	7.4541
9	change	109	7.3009
10	tell	119	5.3562

adjectives

	collocate	cofreq	tscore
1	first	8046	82.7350
2	same	7499	81.7422
3	long	4383	62.0985
4	last	2691	42.3590
5	short	1222	31.2949
6	several	1059	27.6135
7	second	1059	25.7717
8	next	1134	25.3035
9	more	1305	25.1709
10	much	832	23.7150

modified nouns

	base	cofreq	tscore
1	limit	364	18.9060
2	period	273	16.0679
3	scale	157	12.3615
4	round	144	11.3547
5	consuming	129	11.3365
6	interval	114	10.5780
7	zone	112	10.3038
8	i	133	9.9356
9	t	82	8.5963
10	deposit	74	8.3752

modifying nouns

	collocate	cofreq	tscore
1	question	209	14.3329
2	reaction	133	11.4606
3	part	137	11.3837
4	lunch	127	11.1795
5	response	125	11.0810
6	dinner	123	10.9219
7	leisure	113	10.3553
8	transit	101	9.9933
9	night	95	9.2398
10	journey	83	9.0547

Figure 23: Presentation of collocates in *Treebank.info*

6. Conclusion

We have seen in the present paper that collocation candidate extraction is a complex task that has been considerably improved in the course of the past twenty years or so. Nonetheless, the candidate lists generated by any current system are far from perfect. We shall briefly review the achievements but also challenges for future work in this section.

6.1 Achievements

There are two main points to stress again at the end of the present paper. First, there is a wide range of parameters to tune in order to get optimal results from automatic collocation candidate extraction for use in lexicography. Lexicographers have to be aware of the influence of these parameters in order to take sensible decisions for their lexicographic project.

Secondly, and possibly more importantly, we have shown that extracting collocation candidates from a dependency-parsed corpus offers significant improvements over the window-based approach and also outperforms the PoS-pattern approach in most use cases. The improvement lies in the quality of the retrieved lists, so interestingly not only precision but also recall values are quite high even though the lists generated by the approach are always among the shortest candidate lists that can sensibly be extracted from the corpus. With the integration of such a collocation candidate extraction tool into our existing *Treebank.info* project, we can make freely available the power of this methodology to researchers who would not want to go through all the technical hassle still needed to parse corpora and to extract meaningful results from the parsed version. The wide range of configuration options would even allow one principal investigator to set the settings for a certain collocation database according to the needs of the respective project and then have lexicographers work with the pre-set views generated.[48]

In other words, the methodology and the online tool presented here fulfil most of the requirements suggested by Heid et al: "Moreover, it seems necessary to be able to parametrise web services and to allow users to set parameters, before the processing chain is entered" (Heid et al. 2010, 3220). For the task of collocation candidate extraction they go on to list "a few of the parameters we envisage users may wish to set [...]: which grammar to use for parsing, which syntactic type or types of collocations to extract, which association measure(s) to use, how to package (e.g. by syntactic type) and how to sort and lexicographically display the results" (Heid et al. 2010, 3220).

In *Treebank.info*, users can choose which dependency parser to use[49] and they can decide on the syntactic types of collocations before the creation of the database. Association measure and co-frequency threshold can be set during querying and users can exert influence on the presentation of the results by specifying sorting criteria and setting the number of collocates to be displayed.

Of course, the method presented here and available via *Treebank.info* cannot replace a lexicographer, but it can make his/her job easier. There is still much work to be done manually in order to create a collocations dictionary, but better tools may lead to better results or may simply be less time-consuming.[50]

[48] To the best of our knowledge, the *Sketch Engine* also offers such features commercially for use with a PoS pattern approach.

[49] The figure given by Heid et al. (2010, 3221) also lists other types of grammar, such as an LFG, though, whereas *Treebank.info*'s architecture is currently limited to dependency models.

[50] There is a range of features that are important for the lexicographic workflow independently of the extraction method chosen. Thus, the integration into a dictionary writing system or the promotion of good examples to the top of the concordance will speed up the creation of dictionary entries. *Treebank.info* in its current state does not offer all the features of such a one-stop solution since it is first and foremost a tool for linguistic analysis.

6.2 Future research

There is a variety of aspects to collocation candidate extraction one may be able to improve in the context of collocation extraction from parsed corpora, besides obvious extensions such as the addition of new languages.[51]

First, the collocation candidate extraction from an automatically parsed corpus is always limited by the parser's accuracy. Given that the Stanford Parser was used for our evaluation here, the results presented by Cer et al. (2010) indicate that there may be room for improvement with a more accurate parser. Even with the same parser, one sometimes has the choice of different grammatical models which may be better or worse for the collocations needed, so a future study may look at these factors and compare the results against the gold standard again.

Secondly, the ranking of "good examples" to the top of the concordance (as proposed by Kilgarriff et al. (2008) and available in the *Sketch Engine*) may benefit from the use of parsed corpora, since these permit to restrict the syntactic complexity more accurately than the application of mere word counts.

A third and more challenging point is to attempt automatic semantic groupings of collocation candidates. The application of distributional semantic models (DSM), which again would probably benefit from parsed corpora, appears to be a promising line of research in this respect since it relies on the idea by Harris that "difference of meaning correlates with difference of distribution" (Harris 1954, 156). In other words, we would expect semantically similar words to be found in similar contexts. While such analyses may be computationally expensive, they may significantly speed up the process of reading through a list of collocation candidates and writing a dictionary entry. Rychlý/Kilgarriff (2007) present a computationally less expensive heuristic solution to the problem, which is also implemented in the *Sketch Engine*. Given the mixed results of the process, it may be more successful to use *WordNet* (Fellbaum 1998) or a thesaurus for the semantic grouping of collocates, but the downside of such an approach is that it loses its language-independence and has to rely on the availability of appropriate resources.

Finally, if we look at the lists of collocation candidates generated by our system (and by others), we find that the most strongly associated items are often highly frequent, at least for highly frequent words. While these are of course important, we have to keep in mind the fact that many interesting collocations are not frequent at all, as mentioned in the citation by Lea in section 2.4. Hausmann, citing studies that rely on much smaller corpora, also finds that "[v]iele Kollokationen sind nicht frequent, aber dennoch verfügbar"[52] (Hausmann 1985, 124). With ever-increasing corpus size, the difficulty is not so much that these uses do not occur in the corpus, but they are hidden among a huge number of equally rare but entirely uninteresting co-occurrences, so one of the most important challenges for future work is to find low-frequency collocations as well and reliably separate them from other co-occurrences. Here, a combination of association measures, analyses of the co-occurrence

[51] In the short term, *Treebank.info* will be extended to offer the methodology for German, too.

[52] "Many collocations are not frequent, but nonetheless available", where "available" should be interpreted as mentally available to the native speaker.

with other items, and possibly DSM methods[53] may be able to improve the results.

To sum up we can state that collocation candidate extraction from parsed corpora is one step on the road to perfect lists of collocation candidates, but we still need to refine collocation candidate extraction methods considerably in order to get anywhere near that target.

7. References

Ambati/Reddy/Kilgarriff 2012 = Ambati, Bharat Ram / Reddy, Siva / Kilgarriff, Adam: Word Sketches for Turkish. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). Istanbul, 2012, 2945–2950.

Anthony 2007 = Anthony, Laurence: AntConc (Version 3.2.1w) [Computer Software]. Tokyo, Japan: Waseda University, 2007. Available from <http://www.antlab.sci.waseda.ac.jp/>.

Bartsch 2004 = Bartsch, Sabine: Structural and Functional Properties of Collocations in English: A Corpus Study of Lexical and Pragmatic Constraints on Lexical Co-occurrence. Tübingen: Narr, 2004.

BBI = The BBI Combinatory Dictionary of English. A Guide to Word Combinations. Amsterdam / Philadelphia: Benjamins, 1986. [Third edition 2010]

BNC = The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium. 2007. Details available at <http://www.natcorp.ox.ac.uk/>.

BNCweb = Hoffmann, Sebastian / Evert, Stefan: BNCweb (CQP-Edition, version 4.2). 2009. Details and access available at <http://www.bncweb.info>.

Cer et al. 2010 = Cer, Daniel / de Marneffe, Marie-Catherine / Jurafsky, Daniel / Manning, Christopher D.: Parsing to Stanford Dependencies: Trade-offs between speed and accuracy. In: Proceedings of LREC 2010. Malta, 2010, 1628–1632.

Church / Hanks 1990 = Church, Kenneth W. / Hanks, Patrick: Word Association Norms, Mutual Information, and Lexicography. In: Computational Linguistics 16/1. 1990, 22–29.

COBUILD1 = Collins Cobuild English Language Dictionary. London, 1987.

Cowie 1981 = Cowie, Anthony P.: The Treatment of Collocations and Idioms in Learners' Dictionaries. In: Applied Linguistics II/3. 1981, 223–235.

Crystal 2008 = Crystal, David: A Dictionary of Linguistics and Phonetics, Sixth Edition. Oxford: Blackwell, 2008.

Daille 1994 = Daille, Béatrice: Approche mixte pour l'extraction automatique de terminologie: statistiques lexicales et filtres linguistiques. PhD thesis, Université Paris 7, 1994.

Dunning 1993 = Dunning, Ted E. Accurate methods for the statistics of surprise and coincidence. In: Computational Linguistics 19/1. 1993, 61–74.

Evert 2005 = Evert, Stefan: The Statistics of Word Cooccurrences: Word Pairs and Collocations. Stuttgart: Institut für maschinelle Sprachverarbeitung, 2005.

Evert 2009 = Evert, Stefan: Corpora and Collocations. In: Lüdeling, Anke / Kytö, Merja (edd.): Corpus Linguistics: An International Handbook, Vol. 2. Berlin: Mouton de Gruyter, 2009, 1212–1248.

[53] Katz/Giesbrecht (2006) show that Latent Semantic Analysis can be used to calculate the degree of compositionality of a multi-word expression, at least to a certain extent. The accuracy achieved by their method is probably not sufficient for use in a lexicographic project, but since they state that "[c]ertainly some recognition of the syntactic structure would improve results" (2006, 17), there is hope that using parsed corpora may indeed make the method usable for the purpose of identifying interesting collocations while discarding fully idiomatic and entirely free combinations at the extreme ends of a compositionality scale.

Evert / Krenn 2005 = Evert, Stefan / Krenn, Brigitte: Using small random samples for the manual evaluation of statistical association measures. In: Computer Speech and Language 19. 2005, 450–466.

Fellbaum 1998 = Fellbaum, Christiane (ed.): Wordnet: An Electronic Lexical Database. Cambridge, MA: MIT Press, 1998.

Francis 1965 = Francis, W. Nelson: A Standard Corpus of Edited Present-Day American English. In: College English 26/4. 1965, 267–273.

Firth 1957/1968 = Firth, John Rupert: A synopsis of linguistic theory: 1930–55. In: Palmer, Frank R. (ed.): Selected Papers by J. R. Firth 1952–59. London/Harlow: Longman. 1968, 168–205. [First published in 1957]

Götz-Votteler / Herbst 2009 = Götz-Votteler, Katrin / Herbst, Thomas: Innovation in advanced learner's dictionaries of English. In: Lexicographica 25. 2009, 47–66.

Halliday / Hasan 1976 = Halliday, Michael A. K. / Hasan, Ruqaiya: Cohesion in English. London: Longman, 1976.

Harris 1954 = Harris, Zelig: Distributional structure. In: Word 10, 1954, 146–162.

Hausmann 1979 = Hausmann, Franz Josef: Un dictionnaire de collocations est-il possible? In: Travaux de linguistique et de littérature 17. 1979, 187–195.

Hausmann 1984 = Hausmann, Franz Josef: Wortschatzlernen ist Kollokationslernen. In: Praxis des neusprachlichen Unterrichts 31. 1984, 395–406.

Hausmann 1985 = Hausmann, Franz Josef: Kollokationen im deutschen Wörterbuch: Ein Beitrag zur Theorie des lexikographischen Beispiels. In: Bergenholtz, Henning / Mugdan, Joachim (edd.): Lexikographie und Grammatik. Tübingen: Niemeyer. 1985, 118–129.

Hausmann 2004 = Hausmann, Franz Josef: Was sind eigentlich Kollokationen? In: Steyer, Kathrin: Wortverbindungen – mehr oder weniger fest. Berlin / New York: de Gruyter. 2004, 309–334.

Hausmann 2008 = Hausmann, Franz Josef: Kollokationen und darüber hinaus: Einleitung in den thematischen Teil „Kollokationen in der europäischen Lexikographie und Wörterbuchforschung". In: Lexicographica 24. 2008, 1–8.

Heid 1998 = Heid, Ulrich: Towards a corpus-based dictionary of German noun-verb collocations. In: Fontenelle, Thierry / Hiligsmann, Philippe / Michiels, Archibald / Moulin, André / Theissen, Siegfried (edd.): EURALEX'98 Proceedings: Papers submitted to the Eighth EURALEX International Congress on Lexicography in Liège, Belgium. Liège, University of Liège, English and Dutch Departments, 1998, 301–312.

Heid et al. 2008 = Heid, Ulrich / Fritzinger, Fabienne / Hauptmann, Susanne / Weidenkaff, Julia / Weller, Marion: Providing corpus data for a dictionary for German juridical phraseology. In: Storrer, Angelika / Geyken, Alexander / Siebert, Alexander / Würzner, Kay-Michael (edd.): Text Resources and Lexical Knowledge: Selected Papers from the 9th Conference on Natural Language Processing KONVENS 2008. Berlin / New York: Mouton de Gruyter, 2008, 131–144.

Heid et al. 2010 = Heid, Ulrich / Fritzinger, Fabienne / Hinrichs, Erhard / Hinrichs, Marie / Zastrow, Thomas: Term and collocation extraction by means of complex linguistic web services. In: Proceedings of LREC 2010. Malta, 2010, 3215–3221.

Herbst 1996 = Herbst, Thomas: What are collocations: *sandy beaches* or *false teeth*? In: English Studies 77/4, 1996, 379–393.

Herbst 2011 = Herbst, Thomas: Choosing sandy beaches – collocations, probabemes and the idiom principle. In: Herbst, Thomas / Faulhaber, Susen / Uhrig, Peter (edd.): The Phraseological View of Language: A Tribute to John Sinclair. Berlin / Boston: de Gruyter Mouton. 2011, 27–57.

Herbst / Klotz 2009 = Herbst, Thomas / Klotz, Michael: Syntagmatic and Phraseological Dictionaries. In: Cowie, Anthony P. (ed.): The Oxford History of English Lexicography. Oxford: Oxford University Press, 2009, 219–244.

Herbst / Mittmann 2008 = Herbst, Thomas / Mittmann, Brigitta: Collocation in English dictionaries at the beginning of the twenty-first century. In: Lexicographica 24. 2008, 103–119.

Hoffmann et al. 2008 = Hoffmann, Sebastian / Evert, Stefan / Smith, Nicholas / Lee, David / Berglund Prytz, Ylva: Corpus Linguistics with *BNCweb*: A Practical Guide. Frankfurt: Peter Lang, 2008.

Ivanova et al. 2008 = Ivanova, Kremena / Heid, Ulrich / Schulte im Walde, Sabine / Kilgarriff, Adam / Pomikálek, Jan: Evaluating a German Sketch Grammar: A Case Study on Noun Phrase Case. In: LREC 2008. Marrakech, 2008, 2101–2107.

Katz / Giesbrecht 2006 = Katz, Graham / Giesbrecht, Eugenie: Automatic Identification of Non-Compositional Multi-Word Expressions using Latent Semantic Analysis. In: Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties. Sidney: Association for Computational Linguistics. 2006, 12–19.

Kilgarriff et al. 2004 = Kilgarriff, Adam / Rychlý, Pavel / Smrz, Pavel / Tugwell, David: The Sketch Engine. In: Williams, Geoffrey / Vessier, Sandra (edd.): Proceedings of the Eleventh EURALEX International Congress. Lorient, France: Faculté des Lettres es Sciences Humaines, Université de Bretagne Sud, 2004, 105–111. Details and access available at <http://www.sketchengine.co.uk>.

Kilgarriff et al. 2008 = Kilgarriff, Adam / Husák, Miloš / McAdam, Katy / Rundell, Michael / Rychlý, Pavel: GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In: Bernal, Elisenda / DeCesaris, Janet (edd.): Proceedings of the XIII EURALEX International Congress. Barcelona: Universitat Pompeu Fabra, 2008, 425–433.

Kilgarriff et al. 2010 = Kilgarriff, Adam / Kovář, Vojtěch / Krek, Simon / Srdanović, Irena / Tiberius, Carole: A Quantitative Evaluation of Word Sketches. In: Dykstra, Anne / Schoonheim, Tanneke (edd.): Proceedings of the XIV Euralex International Congress. Leeuwarden: Fryske Akademy, 2010, 372–379.

Kjellmer 1994 = Kjellmer, Göran: A Dictionary of English Collocations. Oxford: Clarendon, 1994.

Klein / Manning 2003 = Klein, Dan / Manning, Christopher D.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Sapporo: Association for Computational Linguistics. 2003, 423–430.

Klotz 2000 = Klotz, Michael: Grammatik und Lexik: Studien zur Syntagmatik englischer Verben. Tübingen: Stauffenburg, 2000.

Knowles / Zuraidah 2008 = Knowles, Gerry / Zuraidah, Mohd Don: The notion of a “lemma”: Headwords, roots and lexical sets. In: International Journal of Corpus Linguistics 9/1. 2004, 69–81.

Lea 2007 = Lea, Diana: Making a Collocations Dictionary. In: Götz-Votteler, Katrin / Herbst, Thomas (edd.): Collocation and Creativity, Zeitschrift für Anglistik und Amerikanistik 55/3, 261–272.

Leech et al. 1994 = Leech, Geoffrey / Garside, Roger / Bryant, Michael: CLAWS4: The tagging of the British National Corpus. In: Proceedings of the 15th International Conference on Computational Linguistics (COLING 94), Kyoto, 1994, 622–628.

Lehmann / Schneider 2011 = Lehmann, Hans Martin / Schneider, Gerold: A large-scale investigation of verb-attached prepositional phrases. In: Rayson, Paul / Hoffmann, Sebastian / Leech, Geoffrey (edd.): Methodological and Historical Dimensions of Corpus Linguistics. Helsinki: Research Unit for Variation, Contacts, and Change in English. Available online at <http://www.helsinki.fi/varieng/journal/volumes/06/lehmann_schneider/>, retrieved 10 May 2012.

Marcus et al. 1993 = Marcus, Mitchell P. / Santorini, Beatrice / Marcinkiewicz, Mary Ann: Building a Large Annotated Corpus of English: The Penn Treebank. In: Computational Linguistics, 19/2. 1993, 313–330.

de Marneffe / Manning 2008 = de Marneffe, Marie-Catherine / Manning, Christopher D.: The Stanford typed dependencies representation. In: Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation. Manchester, 2008.

MCD = Macmillan Collocations Dictionary for Learners of English. Oxford, 2010.

Nesselhauf 2005 = Nesselhauf, Nadja: Collocations in a Learner Corpus. Amsterdam / Philadelphia: Benjamins, 2005.

OCD1 = Oxford Collocations Dictionary for Students of English. Oxford, 2002.

OCD2 = Oxford Collocations Dictionary for Students of English, Second Edition. Oxford, 2009.

Proisl / Uhrig 2012 = Proisl, Thomas / Uhrig, Peter: Efficient Dependency Graph Matching with the IMS Open Corpus Workbench. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12). Istanbul, 2012, 2750–2756.

Rundell 2011 = Rundell, Michael: How Dictionaries are Written: Macmillan Collocations Dictionary. 2011. Available at <http://www.macmillandictionaries.com/features/how-dictionaries-are-written/macmillan-collocations-dictionary/>, retrieved 10 May 2012.

Russell / Norvig 2010 = Russell, Stuart J. / Norvig, Peter: Artificial Intelligence: A modern approach. Third edition. Upper Saddle River, NJ: Pearson, 2010.

Rychlý / Kilgarriff 2007 = Rychlý, Pavel / Kilgarriff, Adam: 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: Companion Volume Proceedings of the Demo and Poster Sessions. Prague: Association for Computational Linguistics. 2007, 41–44.

Schmid 1994 = Schmid, Helmut: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: Proceedings of International Conference on New Methods in Language Processing, Manchester, 1994.

Schmid 2008 = Schmid, Helmut: Tokenizing and part-of-speech tagging. In: Lüdeling, Anke / Kytö, Merja (edd.): Corpus Linguistics: An International Handbook, Vol. 1. Berlin: Mouton de Gruyter, 2008, 527–551.

Seretan 2011 = Seretan, Violeta: Syntax-Based Collocation Extraction. Dordrecht: Springer, 2011.

Siepmann 2005 = Siepmann, Dirk: Collocation, Colligation and Encoding Dictionaries. Part I: Lexicological Aspects. In: International Journal of Lexicography 18/4. 2005, 409–443.

Sinclair 1991 = Sinclair, John McH.: Corpus Concordance Collocation. Oxford: Oxford University Press, 1991.

Smadja 1993 = Smadja, Frank: Retrieving collocations from text: Xtract. In: Computational Linguistics 19/1. 1993, 143–177.

Tarp 2008 = Tarp, Sven: Lexicography in the Borderland between Knowledge and Non-Knowledge. Tübingen: Niemeyer, 2008.

Tognini-Bonelli 2001 = Tognini-Bonelli, Elena: Corpus Linguistics at Work. Amsterdam / Philadelphia: Benjamins, 2001.

Treebank.info = Uhrig, Peter / Proisl, Thomas: Treebank.info. 2011. Details and access available at <http://treebank.info>.