Chapter 6
Collocation Candidate Extraction from Dependency-Annotated Corpora: Exploring Differences across Parsers and Dependency Annotation Schemes

Peter Uhrig, Stefan Evert, and Thomas Proisl

Abstract Collocation candidate extraction from dependency-annotated corpora has become more and more mainstream in collocation research over the past years. In most studies, however, the results of one parser are compared to those of relatively "dumb" window-based approaches only. To date, the impact of the parser used and its parsing scheme has not been studied systematically to the best of our knowledge. This chapter evaluates a total of 8 parsers on 2 corpora with 20 different association measures plus several frequency thresholds for 6 different types of collocations against the *Oxford Collocations Dictionary for Students of English* (2nd edition; 2009). We find that the parser and parsing scheme both play a role in the quality of the collocation candidate extraction. The performance of different parsers can differ substantially across different collocation types. The filters used to extract different types of collocations from the corpora also play an important role in the trade-off between precision and recall we can observe. Furthermore, we find that carefully sampled and balanced corpora (such as the BNC) seem to have considerable advantages in precision, but of course for total coverage, larger, less balanced corpora (such as the web corpus used in this study) take the lead. Overall, log-likelihood is the best association measure, but for some specific types of collocation (such as adjective-noun or verb-adverb), other measures perform even better.

P. Uhrig (✉) · S. Evert · T. Proisl
Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
e-mail: peter.uhrig@fau.de

P. Cantos-Gómez, M. Almela-Sánchez (eds.), *Lexical Collocation Analysis*,
Quantitative Methods in the Humanities and Social Sciences,
https://doi.org/10.1007/978-3-319-92582-0_6

1 Introduction

While it is common practice to start a chapter on collocation candidate extraction with a lengthy discussion of the various concepts of collocation, we will keep this discourse to a minimum:[1] For the purpose of this paper, we define collocation as the combination of two lexical items as listed in collocations dictionaries, in our case in the *Oxford Collocations Dictionary for Students of English* (2nd edition; 2009). The rationale behind this is that the present paper aims to determine the best strategy to create lists of collocation candidates that can then be used in lexicography.

Evert (2004) identifies three approaches to the extraction of collocation candidates: segment-based co-occurrences, distance-based co-occurrences and relational co-occurrences. The segment-based approach relies on the statistical analysis of words that co-occur within some segment of text, e.g. a sentence or paragraph. The distance-based approach analyses words that co-occur within a short distance from each other that is usually defined as a window of orthographic words. Those two approaches require very little preprocessing and therefore were very popular when sufficiently fast and robust syntactic parsers were not readily available. The third approach, relational co-occurrences, analyses co-occurrences of words that are related by some (usually syntactic) relation. As such, it requires syntactically annotated corpora where the syntactic relation between words is made explicit. This requirement is met by dependency grammar. Studies have shown that relational co-occurrences are generally superior to segment-based or distance-based co-occurrences (cf. Uhrig and Proisl (2012), Bartsch and Evert (2014)).

However, a wide range of dependency parsers are available, and while there are many studies that have worked with such parsers to extract collocation candidates from corpora, their typical approach is to compare the results from one parser with distance-based or segment-based approaches. To date, no study we are aware of systematically compares different parsers against each other to determine the influence of the parser and/or its parsing scheme onto the quality of the extracted data. The present chapter tries to fill this gap.

2 Related Work

With the advent of sufficiently fast and accurate parsers, the extraction of collocation candidates based on syntactic relations, i.e. relational co-occurrences, has become one of the most popular approaches to collocation candidate extraction. All types of syntactic analysis have been used for collocation candidate extraction: partial or shallow syntactic analyses, phrase structure and dependency analyses.

[1] See Bartsch (2004: 27–39, 58–78) for a detailed overview.

Partial or shallow syntactic analyses have been used, for example, by Church et al. (1989), Basili et al. (1994), Kermes and Heid (2003) and Wermter and Hahn (2006). For several languages, the Sketch Engine (Kilgarriff et al. 2004) uses shallow analyses based on regular expressions over part-of-speech tags to define grammatical relations for word sketches. However, shallow parsing strategies have certain limitations. Ivanova et al. (2008), for example, find that for German the shallow approach is inferior to richer parsing strategies.

Phrase structure analyses have been used, for example, by Blaheta and Johnson (2001), Schulte im Walde (2003), Zinsmeister and Heid (2003, 2004), Villada Moirón (Villada and Begoña 2005), Seretan (2008) (cf. also Nerima et al. (2003), Seretan et al. (2003, 2004) and Seretan and Wehrli (2006)) and Sangati and van Cranenburgh (2015). It is worth noting that despite using a phrase structure parser, Seretan's extraction is based on grammatical relations between individual words, some of which are explicit in the parser's output, while others have to be inferred from the constituent structure.

Dependency analyses have been used, for example, by Teufel and Grefenstette (1995), Lin (1998, 1999), Pearce (2001), Lü and Zhou (2004), Heid et al. (2008), Weller and Heid (2010), Uhrig and Proisl (2012), Ambati et al. (2012) and Bartsch and Evert (2014).

Covarying collexeme analysis (Gries and Stefanowitsch 2004; Stefanowitsch and Gries 2005) is a minor extension of relational co-occurrences. Instead of analyzing words that are connected by a dependency relation, i.e. words that occur in two different slots in the same dependency relation, it analyses "words occurring in two different slots in the same construction" (Stefanowitsch and Gries 2009: 942). This means that covarying collexeme analysis introduces a slightly more general notion of co-occurrence: co-occurrence via a more complex syntactic structure instead of co-occurrence via a single dependency relation.

The conventional approach to collocation candidate extraction is to collect co-occurrence data and then rank candidate word pairs according to a measure of statistical association between the words. Such association measures compute a score from the co-occurrence frequency of the word pair and the marginal frequencies of the individual words, usually collected in the form of a 2×2 contingency table. A large number of association measures have been proposed in the literature. Evert (2004: 75–91) thoroughly discusses more than 30 different measures, Pecina (2005) gives a list of 84 measures, 57 of which are based on 2×2 contingency tables, and Wiechmann (2008: 253) compares 47 measures "in a task of predicting human behavior in an eye-tracking experiment". There is also a variety of approaches to the quantitative and qualitative evaluation of association measures for a given purpose, for example, Evert and Krenn (2001), Pearce (2002), Pecina (2005), Pecina and Schlesinger (2006), Wermter and Hahn (2006), Pecina (2010), Uhrig and Proisl (2012), Kilgarriff et al. (2014) and Evert et al. (2017).

Recent work has often focussed on the identification of particular types of lexicalized multiword expressions and complements association measures with other automatic methods for determining, for example, the compositionality (Katz and Giesbrecht 2006; Kiela and Clark 2013; Yazdani et al. 2015), non-modifiability

(Nissim and Zaninello 2013; Squillante 2014) or non-substitutability (Pearce 2001; Farahmand and Henderson 2016) of word combinations. There are also approaches that combine multiple sources of information with machine learning techniques (e.g. Tsvetkov and Wintner 2014). Finally, the approach taken by Rodríguez-Fernández et al. relies solely on distributional methods for a "semantics-driven recognition of collocations" (Rodríguez-Fernández et al. 2016: 499).

3 Methodology

3.1 Corpora

We evaluated the collocation candidate extraction from two very different corpora. The first is the *British National Corpus* (BNC) compiled in the early 1990s and comprising roughly 100 million words of running text. The BNC is carefully sampled to contain a wide range of text types, including 10 per cent spoken text. Since, by modern standards, the BNC cannot be counted among large corpora anymore, and since it is considerably older than the latest edition of the dictionary we use as gold standard (see Sect. 3.4), and since it is much smaller than what the compilers of the dictionary used, we decided to include ENCOW16A (Schäfer/Bildhauer Schäfer and Bildhauer 2012, Schäfer 2015), a corpus of English web pages comprising 16.8 billion tokens according to the official corpus documentation. Since we skipped all words that were recognized as so-called boilerplate (e.g. website navigation) by the COW team's software, the actual size of the corpus used in the present study is roughly 12.1 billion tokens.

3.2 Models and Parsers

For parsing to English phrase structure trees, there is only one basic standard, the Penn Treebank style (see Marcus et al. 1993). For English Dependencies, there exist different (often similar but not identical) styles, although much of the recent research seems to converge in the direction of Universal Dependencies (see Sect. 3.2.5 below). Since the decisions taken in the design of a dependency model are likely to influence the accuracy of collocation candidate extraction based on direct relations, we evaluate a set of five models, which are described briefly below together with the parsers that use them.

3.2.1 Combinatory Categorial Grammar (C&C)

The grammatical model used by C&C (Clark and Curran 2007)[2] is Combinatory Categorial Grammar (CCG; Steedman 2000). The dependency representation takes the form of predicate-argument structures with the predicate describing the relation and the governor and the dependent as arguments. However, C&C's output is the only one that incorporates additional arguments – besides governor and dependent – to cover extra information, for instance, on controlling verbs or on passives.

Thus, in example (1), we can observe that the third argument of the ncsubj predicate is empty ("_"). The dobj predicate only has two arguments.

(1) She considers the minister competent.

(ncsubj considered_1 She_0 _)
(dobj considered_1 minister_3)

In the output for (2) on the other hand, the third argument of the ncsubj predicate is "obj", indicating that while syntactically the element is a subject in this passive sentence, it corresponds to an object of the corresponding active sentence.

(2) The minister was considered competent.

(ncsubj considered_3 minister_1 obj)

For our purpose, grouping active clause object and passive clause subject together makes sense and is in line with the policy adopted by most lexicographers, e.g. in the V-N collocations presented by OCD2 (see Sect. 3.3 below for details). Thus we change the relation from ncsubj to obj in such cases in order to produce what we call "collapsed dependencies". Since the passive subject is ambiguous between direct and indirect object, we also collapse the relations dobj and obj2 to obj for consistency. While this processed C&C output is not fully "off-the-shelf", it has previously been used for collocation identification by Bartsch and Evert (2014) and Evert et al. (2017).

The parsing algorithm of C&C is a custom development "which maximizes the expected recall of dependencies" (Clark and Curran 2007: 495).

3.2.2 LTH (CoNLL 2009; Mate)

Johansson and Nugues (2007) created the dependency model that was used as the basis of the popular shared tasks at the CoNLL conferences from 2007 to 2009:

> "The new format was inspired by annotation practices used in other dependency treebanks with the intention to produce a better interface to further semantic processing than existing

[2] http://www.cl.cam.ac.uk/~sc609/candc-1.00.html

> methods. In particular, we used a richer set of edge labels and introduced links to handle long-distance phenomena such as wh-movement and topicalization." (Johansson and Nugues 2007: 105).

In the meantime the CoNLL shared task has moved towards Universal Dependencies (see Sect. 3.2.5 below), but since mate-tools is not under very active development any more, with the main author working for Google on SyntaxNet now, it still uses the CoNLL 2009 format even in its latest version.

3.2.3 Stanford Typed Dependencies (Malt)

The Stanford Typed Dependencies format is described in detail by de Marneffe and Manning (2008). This also is a legacy format that has been superseded by Universal Dependencies (see Sect. 3.2.5 below), behind whose development it was certainly a driving force. Nonetheless, the Malt Parser with engmalt.linear-1.7 model that uses the projective stack algorithm described in Nivre (2009)[3] is used in this comparison, and the English language model is still based on a Penn Treebank version that makes use of Stanford Dependencies. It should be noted that Malt offers this model for "users who only want to have a decent robust dependency parser (and who are not interested in experimenting with different parsing algorithms, learning algorithms and feature models)"[4] because the focus of the Malt development is on implementing and comparing parsing algorithms – in its current version 1.9.1, it implements nine different algorithms.

3.2.4 CLEAR Style (nlp4j, spaCy)

Two parsers used here make use of the dependency representation called CLEAR style. The developers envisage it as a kind of synthesis of Stanford Dependencies and the (older) CoNLL style: "The dependency conversion described here takes the Stanford dependency approach as the core structure and integrates the CoNLL dependency approach to add long-distance dependencies, to enrich important relations like object predicates, and to minimize unclassified dependencies." (Choi and Palmer 2012: 6).

The dependency representation was created for ClearNLP (Choi and Palmer 2011; Choi and McCallum 2013) developed by Emory University's NLP group, which was the predecessor to NLP4J[5] 1.1.3 used in the present chapter. CLEAR style was later adopted by spaCy[6] for English, which we use in version 1.9.0 for this evaluation.

[3] http://www.maltparser.org/

[4] http://www.maltparser.org/mco/mco.html

[5] https://emorynlp.github.io/nlp4j/

[6] https://spacy.io/

While we would expect these parsers to produce comparable results, nlp4j does not follow the guidelines of the CLEAR style in the following example, while spaCy does:

(3) She is a competent minister.

Here, we would expect *competent* to be analysed as an adjectival modifier of *minister*, which is what spaCy does:

amod(minister, competent).

However, nlp4j consistently outputs the following relation:

nmod(minister, competent).

This is a nominal modifier, which is inconsistent with nlp4j's own PoS tagging, where *competent* is in fact tagged as an adjective. Parsing the entire BNC, nlp4j did not output a single amod relation. We will see in the evaluation below how this behaviour affects the collocation candidate extraction for noun-adjective collocations.

3.2.5 Universal Dependencies (Stanford, Stanford Converter [OpenNLP], SyntaxNet)

As hinted above, the Universal Dependencies[7] annotation scheme is on the point of becoming the standard for dependency parsing for any language:

> "The general philosophy is to provide a universal inventory of categories and guidelines to facilitate consistent annotation of similar constructions across languages, while allowing language-specific extensions when necessary." (http://universaldependencies.org/introduction.html).

In our comparison, the neural network dependency parser (Chen and Manning 2014) that is part of Stanford CoreNLP (Manning et al. 2014)[8] and Google's SyntaxNet with the Parsey McParseface model (Andor et al. 2016)[9] use Universal Dependencies, however in slightly different versions.[10] While SyntaxNet is limited to the standard "basic dependencies", Stanford's neural network parser can also produce "enhanced dependencies" and "enhanced++ dependencies" (Schuster and Manning 2016). The basic universal dependencies always form a tree (in the computer science sense of the word), i.e. each word is governed by exactly one other word unless it is the root of the sentence. The enhanced and enhanced++ representations "aim[...] to make implicit relations between content words more

[7] http://universaldependencies.org

[8] https://stanfordnlp.github.io/CoreNLP/

[9] https://github.com/tensorflow/models/tree/master/syntaxnet

[10] To date, the following revisions have been released: 1.0, 1.1, 1.2, 1.3, 1.4, 2.0

explicit by adding relations and augmenting relation names" (Schuster and Manning 2016: 2372). The additional relations may break the tree structure and the resulting analyses are (potentially cyclic) directed graphs.

Stanford CoreNLP and the Stanford Parser also include converters for converting a constituency analysis to a basic dependency analysis and for converting from basic dependencies to an enhanced and enhanced++ representation. We use only the former to convert the phrase structure analyses of Apache OpenNLP[11] to basic dependencies. This means that CoreNLP basic and Apache OpenNLP use exactly the same set of Universal Dependencies.

3.2.6 Summary

In sum we compare 11 combinations of parsers and models/postprocessing options in the present study, which are listed in Table 6.1.

3.3 Gold Standard

The gold standard used in the present study, i.e. the reference against which all parsers and models are compared, is the *Oxford Collocations Dictionary for Learners of English*, 2nd edition (OCD2 2009). It was compiled by lexicographers based on corpora consisting of "almost two billion words of text in English taken from up-to-date sources from around the world" (OCD2: vi). To our knowledge, the exact composition of the corpus collection has never been published, although we can assume that the BNC, which is the sole basis of the 1st edition of the dictionary

Table 6.1 Parsers and models/postprocessing options used in the present study

Parser	Model and postprocessing (if applicable)
C&C 1.00	Default
C&C 1.00	Collapsed
Stanford CoreNLP 3.8.0	Dependency neural network; basic dependencies
Stanford CoreNLP 3.8.0	Dependency neural network; enhanced dependencies
Stanford CoreNLP 3.8.0	Dependency neural network; enhanced++ dependencies
mate-tools 3.6.1	CoNLL2009-ST-English-ALL.anna-3.3
Malt 1.9.1	engmalt.linear-1.7.mco
NLP4J 1.1.3	Default
OpenNLP	Stanford CoreNLP 3.8.0 converter to basic dependencies
spaCy 1.9.0	en_core_web_sm
SyntaxNet 0.2 (April 2017)	Parsey McParseface

[11] https://opennlp.apache.org/

(2002), is included. In its microstructure, OCD2 distinguishes the different senses of the headword lemma, i.e. the base, where necessary and then uses "the grammatical construction as structural divisor" (Klotz and Herbst 2016: 228), i.e. it distinguishes the different types of collocations based on the word class and canonical order of base and collocate. The evaluation in this chapter takes into account the major types of collocations, which are listed in Table 6.1.

3.4 Processing Pipeline

The corpora were processed on FAU's high-performance computing systems to massively parallelize the time-consuming parsing process. After parsing, all instances of dependency relations were extracted together with the part-of-speech tags and lemmata of the governor and the dependent. If a parser supplied lemmata (CoreNLP, C&C, NLP4J, mate, Malt), these were used; if not (SyntaxNet, OpenNLP, spaCy), we applied the same rule-based English lemmatizer that was used in Uhrig and Proisl (2012). In order to ensure a fair evaluation against the OCD2 gold standard and to keep the amount of candidate data manageable, dependency pairs were matched against a word list of 42,720 lemmata, consisting of all headwords from the *Oxford Advanced Learner's Dictionary*, 8th edition (OALD8 2010), and all words that occur in OCD2 in one of the types of collocation listed in Table 6.2 (i.e. all headwords and all collocates). In order not to filter too aggressively, both the word form and the lemma of governor and dependent were compared to the word list; if either word form or lemma of both the governor and the dependent matched entries in the word list, the co-occurrence was accepted into the filtered dataset. For nouns, no difference between common nouns and proper nouns was made to include items such as *God* or various political institutions. However, most proper nouns were of course removed by the word list filter since neither dictionary contains many place names, personal names, or similar items.

Table 6.2 Overview of collocation types in our gold standard

Name in OCD	Abbreviation in this study	Pairs extracted from OCD2
[noun lemma] + verb	NVsubj	8979
verb + [noun lemma]	NVobj	36,670
[noun lemma] + adjective	NJ	86,379
[adjective lemma] + adverb	JV	7135
[verb lemma] + adjective	JR	11,625
[verb lemma] + adverb	VR	12,612

We extracted both unfiltered co-occurrence data (all dependency relations) and data filtered specifically for each collocation type.[12] Contingency tables were then compiled as described by Evert (2004: 33–37), using the UCS toolkit implementation.[13]

For the unfiltered data, lemmata were disambiguated by their part-of-speech category (noun, verb, adjective, adverb). We obtained between 9.2 and 17.1 million contingency tables (i.e. candidate lemma pairs) for the BNC and between 132.8 and 296.8 million contingency tables for ENCOW, depending on the parser and postprocessing used.

For the filtered data, we applied the restrictions listed in Table 6.3. We obtained between 24,148 and 1.6 million contingency tables for BNC, and between 274,492 and 20.6 million contingency tables for ENCOW, depending on syntactic relation[14] and parser.

We use the same set of 20 association measures for candidate ranking as Evert et al. (2017), which includes the most popular measures such as log-likelihood (G^2), t-score (t), z-score with Yates's correction (z), Mutual Information (MI), the Dice coefficient (which is used by the Sketch Engine) and ranking by co-occurrence frequency (f). In addition, we include different versions of the recently proposed ΔP measure (Gries 2013) and a conservative statistical estimate of MI ($\mathrm{MI}_{\mathrm{conf}}$; Johnson 1999). Since our focus here is on the comparison of different parsers, we refer to Evert et al. (2017) for a complete listing of the association measures with equations and references.

4 Evaluation

Following the evaluation methodology of Evert and Krenn (2001), we determine the quality of different n-best candidate lists for each candidate set and association ranking. Consider the example of the verb-object relation identified by the NLP4J parser in the BNC. Among the top 1,000 candidates ranked by log-likelihood, there are 801 true positives (TPs), i.e. actual collocations listed in OCD2. This 1,000-best list hence achieves a precision of 80.10%. However, the recall of this list is only 2.18% of the 36,670 object-verb collocations in OCD2. Similarly, a 10,000-best list achieves a precision of 66.50% and recall of 18.13% (with 6,650 TPs), and a 20,000-best list a precision of 56.16% and recall of 30.63% (with 11,232 TPs).

[12] Unfiltered data can be used to maximize recall, since parsers generally are better at predicting that two items should be connected by a dependency relation than they are at predicting what type of dependency relation connects the two. In the technical terms of parser evaluation, this is the difference between unlabelled and labelled attachment.

[13] http://www.collocations.de/software.html

[14] There are relatively few candidate pairs for verb-adjective and adverb-adjective collocations; the largest numbers of pairs are found for noun-verb (both subjects and objects) and noun-adjective collocations.

Table 6.3 Filters used for each type of collocation

Parser	Subj-V	V-Obj	Adj-N	V-Adj	V-Adv	Adv-Adj
C&C default	ncsubj(VB, NN)	dobj(VB, NN), obj2(VB, NN)	ncmod(NN, JJ)	xcomp(VB, JJ)	ncmod(VB, RB), dobj(VB, RB)	ncmod(JJ, RB)
C&C collapsed	subj(VB, NN)	obj(VB, NN)	ncmod(NN, JJ)	xcomp(VB, JJ)	ncmod(VB, RB), obj(VB, RB)	ncmod(JJ, RB)
CoreNLP basic	nsubj(VB, NN), nmod(VB, NN), acl(NN, VB)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN)	xcomp(VB, JJ), advcl(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)
CoreNLP basic v3 (see Sect. 3.2.5 for discussion)	nsubj(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN)	xcomp(VB, JJ), advcl(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)
CoreNLP enhanced	nsubj(VB, NN), nsubj:xsubj(VB, NN), nmod:agent(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN), nsubj:xsubj(JJ, NN)	xcomp(VB, JJ), advcl:as(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)
CoreNLP enhanced++	nsubj(VB, NN), nsubj:xsubj(VB, NN), nmod:agent(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN), nsubj:xsubj(JJ, NN)	xcomp(VB, JJ), advcl:as(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)
Mate	SBJ(VB, NN)	OBJ(VB, NN)	NMOD(NN, JJ)	PRD(VB, JJ), OPRD(VB, JJ)	ADV(VB, RB), OBJ(VB, RB), MNR(VB, RB), TMP(VB, RB)	AMOD(JJ, RB)
Malt	nsubj(VB, NN), infmod(NN, VB)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN)	acomp(VB, JJ), dep(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)

(continued)

Table 6.3 (continued)

Parser	Subj-V	V-Obj	Adj-N	V-Adj	V-Adv	Adv-Adj
NLP4J	nsubj(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	nmod(NN, JJ), nsubj(JJ, NN)	acomp(VB, JJ), xcomp(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)
OpenNLP	nsubj(VB, NN), nmod(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN), compound(JJ, NN)	xcomp(VB, JJ), advcl(VB, JJ), amod(JJ, VB)	advmod(VB, RB)	advmod(JJ, RB)
spaCy	nsubj(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN)	acomp(VB, JJ), xcomp(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)
SyntaxNet	nsubj(VB, NN)	dobj(VB, NN), nsubjpass(VB, NN)	amod(NN, JJ), nsubj(JJ, NN)	acomp(VB, JJ), xcomp(VB, JJ)	advmod(VB, RB)	advmod(JJ, RB)

Each cell contains all relations used. Part-of-speech restrictions on governor and dependent are given in parentheses. The PoS tags used here are the first two letters of Penn Treebank tags to group various tags for the same word class (e.g. singular/plural for nouns) together

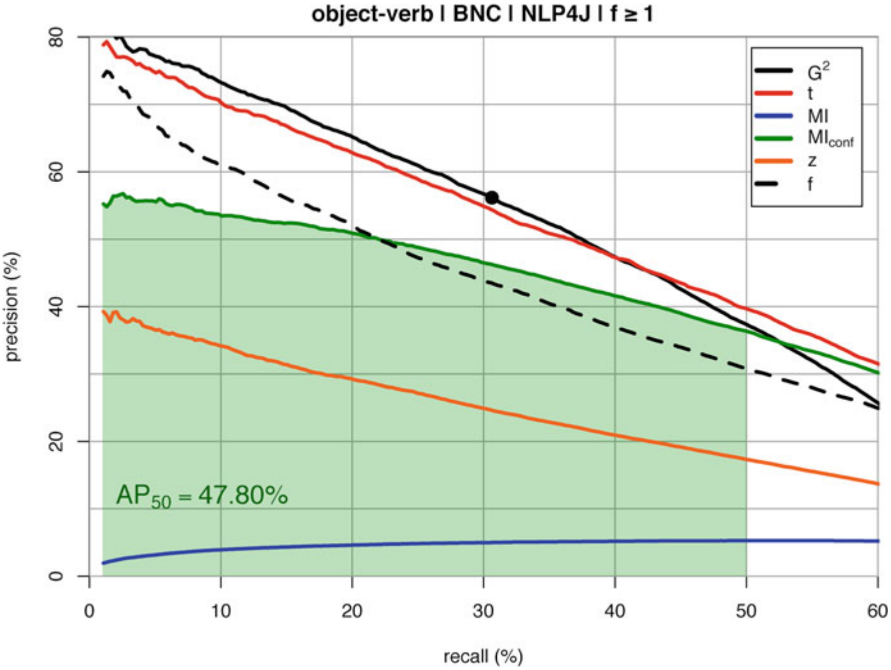

Fig. 6.1 Illustration of evaluation procedure using the methodology of Evert and Krenn (2001) and Evert et al. (2017). (Note that all our plots start at 2% recall since below this value the precision varies wildly and is not very meaningful)

Obviously, the size of an n-best list determines the trade-off between precision and recall. All possible n-best lists can be visualized at a single glance in the form of a precision-recall graph, shown as a solid black line in Fig. 6.1. The 20,000-best list above corresponds to a single point on this line marked by a small dot, at an x-coordinate of 30.63 and a y-coordinate of 56.16. Such precision-recall graphs allow for an easy comparison between different association measures. For example, it is obvious from Fig. 6.1 that log-likelihood (G^2) is a better choice than ranking by co-occurrence frequency (f) because its precision values are always higher at the same recall percentage (mathematicians would say that G^2 is “uniformly better” than f). In turn, f is uniformly better than z-score (z), which is uniformly better than Mutual Information (MI).

Some other cases are less straightforward: G^2 is better than t-score (t) up to 40% recall but worse for higher recall percentages. $\mathrm{MI}_{\mathrm{conf}}$ outperforms co-occurrence frequency for recall above 20% but achieves much lower precision in the front part of the graph. The choice of an optimal association measure thus depends on the recall required by an application. In order to make general comparisons of measures, parsers and other parameters, we need to define a composite evaluation criterion that summarizes the precision/recall graph in a single number. A customary approach is to compute the average of precision values at different recall points, corresponding to the area under a precision/recall graph. The shaded area in Fig. 6.1 illustrates average precision up to 50% recall (AP_{50}) for the MI_{conf} ranking, resulting in a score of $AP_{50} = 47.80\%$. Frequency ranking achieves a slightly better score of $AP_{50} = 49.22\%$ and is thus deemed better in our global evaluation. The cutoff at 50% recall is somewhat arbitrary. It is motivated by the fact that no candidate set achieves complete coverage of the gold standard (i.e. 100% recall) and coverage drops considerably if frequency thresholds are applied. Keep in mind that the coverage of a data set corresponds to the rightmost point of the corresponding precision/recall graphs, i.e. the highest recall value that can be achieved.

In the present study, we generated precision/recall graphs comparing all 20 association measures for each combination of collocation type, corpus, parser and frequency threshold. Concerning the latter, we compare the complete candidate set ($f \geq 1$, cf. Figure 6.1) with two different versions of setting a frequency threshold: (i) a threshold based on absolute co-occurrence frequency ($f \geq 5$) can be motivated by statistical considerations (Evert 2004: 133); (ii) a threshold based on a relative co-occurrence frequency of at least 50 instances per billion words of text ($f \geq 50/G$) affects the BNC and ENCOW data in a similar way. Note that the two thresholds are identical for the 100-million-word BNC. For ENCOW, we set the relative threshold at$f \geq 500$ co-occurrences, assuming a reduced effective size of 10 billion words that takes into account that our parsers extracted fewer instances of dependency relations from the same amount of text than for the BNC.

For each condition, we automatically determined the optimal association measure based on AP_{50} scores. These optimal results are used for global comparisons, but we also report more detailed findings from an inspection of the full precision/recall graphs. We also generated precision/recall graphs comparing different parsers (on the same collocation type, corpus and frequency threshold), using either the same association measure for all parsers or the optimal measure for each individual parser.

5 Results and Discussion

5.1 Association Measures

In order to keep the number of association measures manageable in the detailed discussion below, a selection had to be made from the full set of 20 association measures. As detailed in Sect. 4, for every combination of corpus (BNC, ENCOW16A), co-frequency threshold ($f \geq 1, f \geq 5, f \geq 50/G$), relation (subject-verb, verb-object, adjective-noun, verb-adjective, adjective-adverb, verb-adverb) and parser (see list in Table 6.1), the average precision at 50% recall (AP50) for every association measure was calculated, and the association measure with the highest AP50 was determined (i.e. if 50% recall was reached, which is not always the case when a frequency threshold is applied). Table 6.4 shows how often each association measure was

Table 6.4 Winning association measures at AP50 across relations

Assoc. Measure	NVsubj	NVobj	NJ	JV	JR	VR
log.likelihood	47	69	14	0	18	23
t.score	1	3	22	0	9	0
z.score.corr	12	0	0	0	0	0
frequency	0	0	0	38	0	0
MI4	0	0	0	4	0	0
MI.conf	2	0	0	0	9	36
DP.min	0	0	0	0	0	1

shown as the best measure broken down by relation. As we can see, only a few measures occur in the first position in one of the experiments. For the remainder of this chapter, we will only look at the most successful ones, i.e. frequency (which is of course not really an association measure and is only really relevant for verb-adjective collocations), log-likelihood, t-score and MI$_{\text{conf}}$.

There are some general observations which are true of all relations discussed in Sect. 5.2 and which are thus discussed in this section.

On the BNC, using a frequency threshold with MI$_{\text{conf}}$ has a small positive effect. Overall, results without a frequency threshold are quite similar. On ENCOW, on the other hand, MI$_{\text{conf}}$ without a frequency threshold performs poorly, which is probably due to the fact that ENCOW is several orders of magnitude larger than the BNC.

The extent to which a filter on dependency relations improves precision is dependent on the association measure in our dataset: The precision improves substantially for t-score and log-likelihood but much less so for MI$_{\text{conf}}$. We can illustrate this result with a comparison of the precision/recall curves for verb-adverb collocations in Fig. 6.2.

One further observation that is true of all relations is that the difference between Stanford CoreNLP with the enhanced and the enhanced++ models hardly results in visible differences in any of the graphs analysed, so the cover term *enhanced* will be used for both in the remainder of this chapter.

5.2 Comparison of Parsers by Collocation Type

To determine the performance of the parsers separately for each type of collocation, we analysed 16 graphs for each type, which were the result of combining the following factors: corpus (BNC, ENCOW16A), statistics (t-score, log-likelihood, MI$_{\text{conf}}$, frequency) and frequency threshold ($f \geq 1$ [i.e. no threshold], $f \geq 50/G$ [i.e. $f \geq 5$ for the BNC, $f \geq 500$ for ENCOW16A]). We will start with a detailed case study of subject-verb collocations to illustrate the analysis in detail. Since much of this is relevant to all types of collocation, the discussion of the remaining ones will be much less verbose.

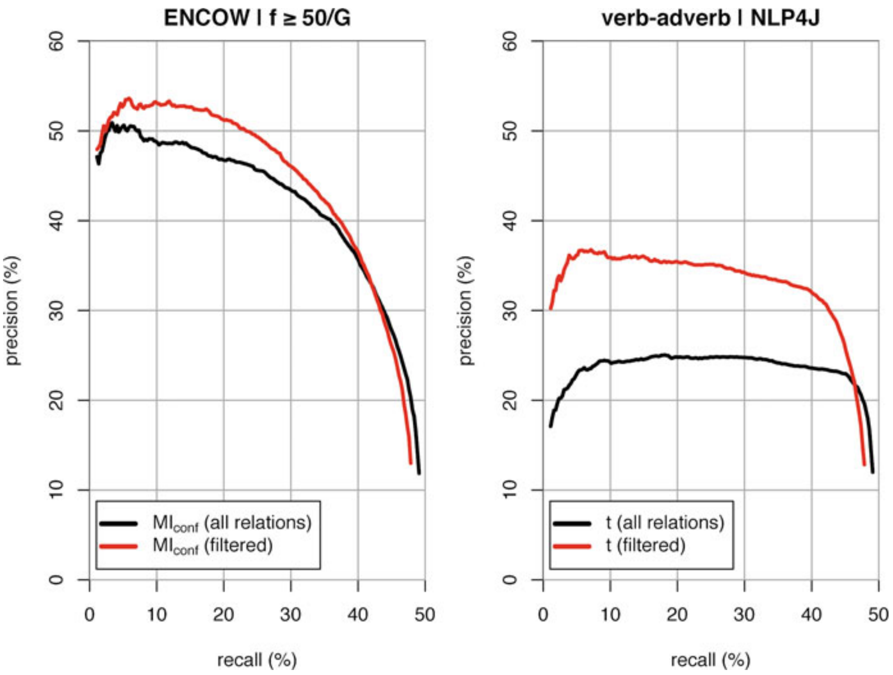

Fig. 6.2 Precision/recall curves for verb-adverb collocations in ENCOW16A with NLP4J

5.2.1 Subject-Verb

Examples:

(4) Her *boss hired* a new secretary.
(5) A new secretary was *hired* by her *boss*.
(6) Her boss wanted to *hire* a new *secretary*.
(7) Her colleague convinced her *boss* to *hire* a new secretary.
(8) Her *boss* had been convinced to *hire* a new secretary.
(9) Her colleague liked the new secretary *hired* by her *boss*.
(10) Her colleague liked the new secretary who had been *hired* by her *boss* the week before.

5.2.2 Overview

For the subject-verb collocations in the BNC, C&C, CoreNLP enhanced and NLP4J form the leading group in terms of precision. The latter only sees straightforward active clause subjects as in example (4) above, whereas C&C and CoreNLP enhanced also take by-agent phrases in the passive (example (5)) and subjects of non-finite subordinate clauses (example (6)) into account.

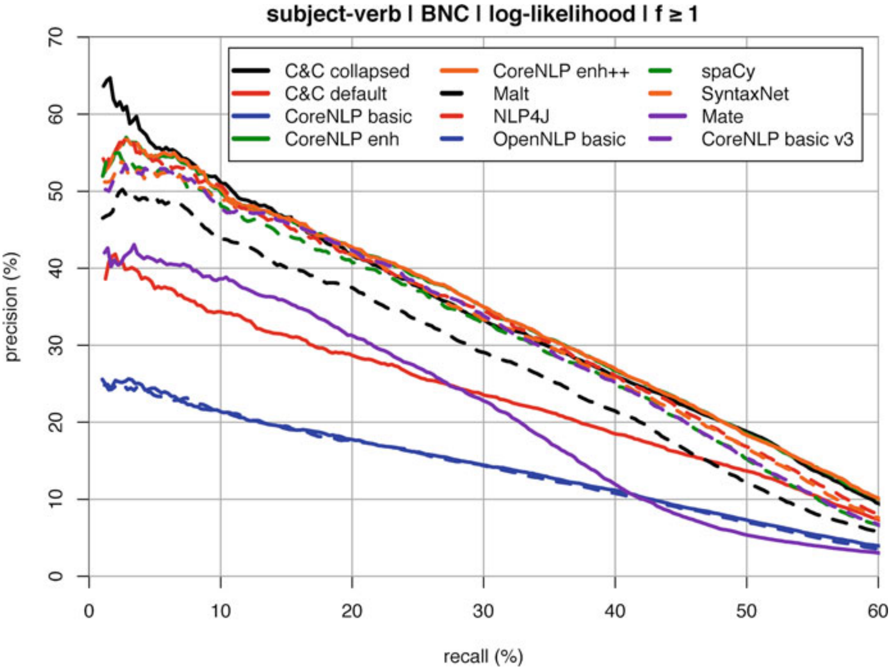

Fig. 6.3 Precision-recall graph for subject-verb collocation candidates from the BNC using log-likelihood and no frequency threshold

In ENCOW16A, CoreNLP basic v3 (see discussion below) performs best without a frequency threshold, but when a frequency threshold of 50/G is applied, recall and precision at above 30% recall are reduced compared to CoreNLP enhanced and C&C, precisely because the latter also include cases such as examples (2) and (3). Surprisingly, mate performs much worse than CoreNLP basic v3, even though it should also show this high precision according to the parsing model. Since precision is generally very low for subject-verb collocations in our experiments on ENCOW16A, a more thorough investigation follows below.

5.2.3 Detailed Discussion

In Fig. 6.3 we can observe that the precision up to 50% recall is very bad for the collocation candidate extraction labelled “Core NLP basic” and very good for the version labelled “CoreNLP basic (v3)”. Both lines in the graph are based on the same output from Stanford CoreNLP, but the collocation candidate extraction is different. This can be explained if we take a look at how CoreNLP processes the example sentences (4) to (10).

Ideally, we would like the parser to find a relation between *boss* and *hire* in all these sentences because all are potential candidates for a subject-verb collocation.[15] However, CoreNLP basic does not recognize such a relation in sentences (6) and (8), whereas CoreNLP enhanced does. Sentence (7) results in a parsing error in CoreNLP, where, in the basic variant, the relation is called acl, which is a clausal modifier of a noun. In CoreNLP enhanced, the relation is specified as acl:to, because the enhanced variant adds the element called “marker” (i.e. the subordinator or infinitive marker) to the relation name. CoreNLP basic is also less explicit than the enhanced variant in the case of the passive *by*-agents in sentences (5), (9), (10), for which the very general nmod (nominal modifier) relation is used, while the enhanced variant uses nmod:agent for (5) and (10) and nmod:by for (9), which probably should also be nmod:agent instead and may thus be due to an error in the conversion rules from basic to enhanced dependencies. In our first run of the collocation candidate extraction, we decided to include both nmod and acl in the extraction rules for subject-verb collocations for CoreNLP basic in order to maximize recall. This, however, led to the extremely bad precision we can witness in Fig. 6.3 (and which is very similar to that of OpenNLP since we also use CoreNLP basic dependencies for it). The curve labelled “CoreNLP basic v3” is geared towards high precision by removing both nmod and acl in the list of possible relations for subject-verb collocations. The curves for CoreNLP enhanced/enhanced++ contain both acl:to and nmod:agent.

For C&C, there is a similar issue in that C&C default does not distinguish between active-clause subjects and passive-clause subjects, which considerably reduces its precision. C&C collapsed, which makes the distinction, is among the top parsers.

Of course, CoreNLP basic v3, SyntaxNet and the other parsers that are at the top of the graphs for some of the association scores might achieve better precisions by sacrificing recall, which cannot be seen from our evaluation plots (up to 50% recall).[16] However, the information is available in the coverage overview plots.

As we can see in Fig. 6.4, the choice really is a trade-off between precision and recall in that CoreNLP basic with all relations finds considerably more relevant items (“true positives”) than CoreNLP basic v3, but at the cost of including a very high number of irrelevant items (“false positives”). When the corpus is large enough and the frequency threshold is relatively low, the differences in coverage are much smaller and high precision becomes the major criterion for the performance of a parser for collocation candidate extraction.

One more observation we can gather from comparing different plots for subject-verb collocation candidates is that precision is on an average level for the BNC (AP50 ∼38.5%) but relatively low for ENCOW16A (AP50 ∼22.5%). This is not an issue of gold standard collocations missing from the corpus, though. Without

[15] That is, of course, if the definition of the collocation type is regarded as a lexical phenomenon with the terminology based on the canonical active-declarative structure.

[16] Except for graphs where the high frequency threshold leads to a coverage of less than 50%

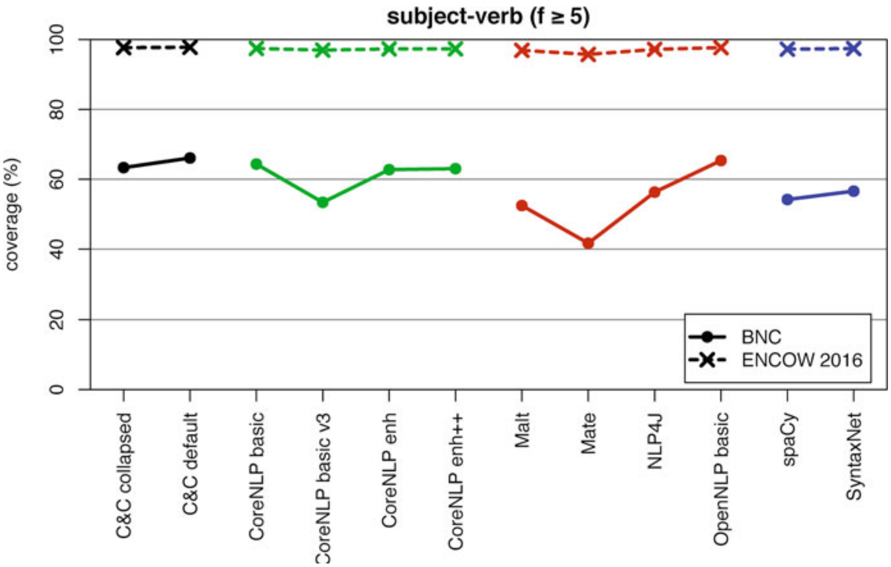

Fig. 6.4 Coverage of subject-verb collocation candidates for BNC and ENCOW 2016 with $f \geq 5$

frequency threshold, coverage is 89.9% for the BNC and 97.9% for ENCOW. For a closer look, we focus on log-likelihood, which achieves good AP50 across both corpora regardless of frequency threshold (justifying coverage without threshold), even though $\mathrm{MI}_{\mathrm{conf}}$ is slightly better on the BNC with $f \geq 5$ (but extremely bad on ENCOW). The plot below shows the full precision-recall curves of log-likelihood (Fig. 6.5):

Thus the problem lies clearly not in a lack of coverage, but in the ranking of candidates, particularly in the case of ENCOW16A. One observation is that coverage is affected very much by frequency threshold, dropping to a bit over 60% (BNC, $f \geq 5$) or even below 50% (ENCOW, $f \geq 50$/G), which suggests that one problem may be that many subject-verb collocations are very infrequent in the two corpora.

In order to determine why ENCOW16A is so much worse than the BNC, the first 1,000 collocation candidates from ENCOW16A (corresponding to a recall of up to 3.17%) and from the BNC (corresponding to a recall of up to 5.81%) were exported for manual inspection for two parsers, CoreNLP enhanced++ and SyntaxNet. Both files overlap, so in total 1,592 pairs were collected for CoreNLP and 1,577 pairs for SyntaxNet. The first 1,000 items from the BNC contain 551 true positives, i.e. items present in the gold standard, for CoreNLP and 522 for SyntaxNet, whereas the first 1,000 items from ENCOW16A only contain 283 true positives for CoreNLP and 285 for SyntaxNet.

The most important reason for the striking difference between the two corpora seems to be repeated usage in ENCOW16A, where the same text appears on many webpages. Often this is boilerplate, as in the following examples:

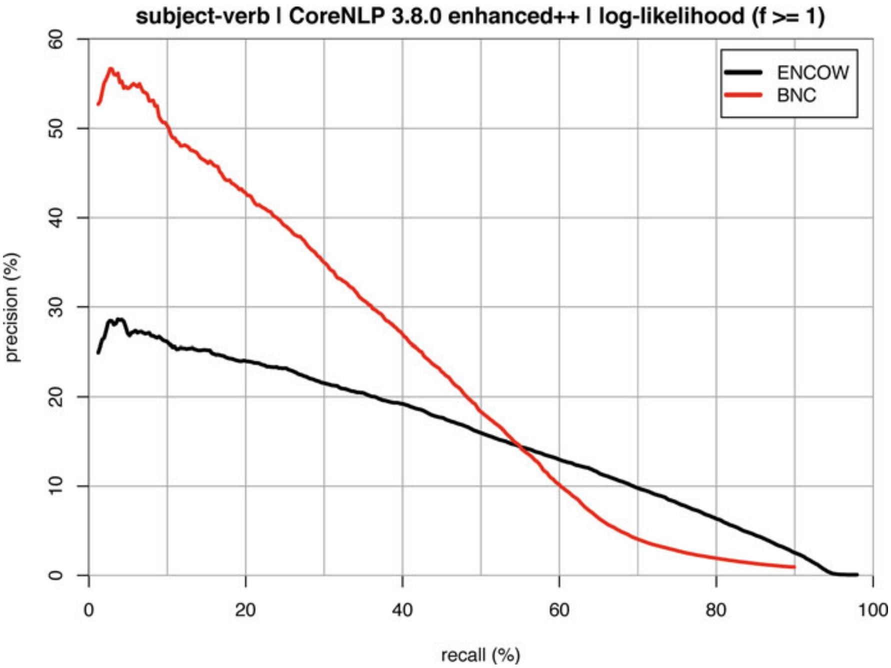

Fig. 6.5 Precision/recall curves for subject-verb collocations with CoreNLP enhanced++, log-likelihood and without a frequency threshold

(11) Grapeshot stores the categories of story you have been exposed to. (>200,000)
(12) Failure to return items with all the required documentation will result in a delay in processing the return and may even invalidate the return itself. (>20,000)
(13) People also look for caravans to rent, apple 3 g iphone, small holdings to rent, top online classifieds for pets in England, laptop computers, bedsits in london, free world ads and many more interesting items. (>26,000)

Sentence (11) can be found on many different websites because Grapeshot is an online marketing company. Sentence (12) is from the return policy of an online shoe store from which more than 20,000 product pages found their way into the corpus. Sentence (13) appears to be search-engine spam, i.e. a set of many webpages whose only purpose is to appear at the top of the search results for many search terms and earn money through ads. With such high frequencies, it is of course not surprising that the combination of *Grapeshot* + *store* takes the second-highest position of all collocation candidates in ENCOW16A for SyntaxNet.[17] Some more such candidates in the top 1,000 in ENCOW16A are *type* + *visit*,

[17] CoreNLP produces a parsing error on this sentence so that *Grapeshot stores* is wrongly analysed as a nominal compound.

widget + *give*, *site* + *function*, *website* + *use*, *site* + *set*, *cookie* + *store*, *list* + *update*, *story* + *match*, *delivery* + *take*, *site* + *use* and *feature* + *require*.

There is one further problem of repeated usage: If the parser produces an error in the parse for this particular sentence, it will do the same in all repeated instances. In sentence (13), *caravan* should be analysed as object of *rent* and thus should not occur in the list in the first place, but it is in fact treated as subject of *rent* by both parsers. This problem is particularly pronounced in sentence fragments with past participles, where the parser often identifies the participle as past tense verb and thus the object in front of it as subject:

(14) All rights reserved. (error only in SyntaxNet)
(15) No pun intended / Pun intended. (error in both parsers)

The combination of *right* + *reserve* is the top subject-verb collocation candidate for ENCOW16A in our list for SyntaxNet, and again it is due to a parsing error combined with completely skewed frequencies.

There are more such cases of repeated fragments, which can be part of completely different texts. For instance, the combination *allah* + *bless* occurs frequently, since it is due to the conventionalized complimentary phrase given in (16), which is attached to the names of prophets in Islam.

(16) may Allah bless him and grant him peace

The combination occurs almost 18,000 times, with the bulk of these hits coming from one website on Islamic topics (bewley.virtualave.net), which, according to its start page, provides mainly transcripts of talks and translations of texts from Arabic. Still, the phrase is added to every occurrence of *Mohammed* or *Messenger of Allah*, so it is no real boilerplate but just convention.[18]

ENCOW16A is of course also skewed in many other respects. As expected in a web corpus, there is some language related to computer technology or innovations that are relevant for computers, although the vocabulary filter will already have eliminated many of these. Examples are *cursor* + *hover*, *screen* + *freeze*, *blog* + *cover* and *administrator* + *accept*.

Furthermore, it is likely that our gold standard, OCD2, is biased towards British English, so collocation candidates from other varieties (in particular US-American English) will also influence the precision negatively, e.g. *congress* + *enact*.

Let us now turn to the reasons why we are still far from 100% precision at the top of the collocation candidate list, even in the BNC.

One reason is the number of co-occurrences with the verb *be*. Out of the 1,592 (CoreNLP enhanced++)/1,577 (SyntaxNet) items in the combined top 1,000 list from ENCOW16A and the BNC, there are 128/162 candidates with the verb *be*, 124/155 of which (113/131 from the BNC, 97/117 from ENCOW16A) are false positives, i.e. are not listed in OCD2. The top 10 of the list from ENCOW16A

[18] The same is true of the alternative form "peace be upon him", which occurs more than 10,000 times but does not propel *peace* + *be* into the to 1,000 collocation candidates.

comprises *way*, *reason*, *problem*, *thing*, *point*, *question*, *aim*, *purpose*, *goal* and *suggestion*. Except for *goal*, these are all quite strong in the BNC, too. It is clear that even if such items co-occur relatively frequently with *be*, it is questionable whether they should be listed in a collocations dictionary. Still, some are of course similar in fixedness and frequency to the seven true positives[19] in the lists, *cause*, *difference*, *focus*, *issue*, *secret*, *time* and *truth*, so what it is that made the lexicographers include them in OCD2 but not *reason*, *problem* or *point* remains an open question.

Another large proportion of false positives are unspecific combinations. Some of these occur with general (pro)nouns, e.g. *anyone* + *know*, *someone* + *tell* or *people* + *want*, but many are just common words occurring more frequently than expected based on their individual frequencies, such as *company* + *pay*, *group* + *meet*, *school* + *have* or *wife* + *die*, a fact that is "neither particularly surprising nor particularly interesting" (Herbst 1996: 382), just like the example of *sell* + *house* quoted by Herbst.

Finally, there are cases that may just as well figure in a collocations dictionary, for instance, *section* + *describe*, *government* + *propose* or *budget* + *grow*, but that are not part of our gold standard.

A complementary perspective is offered by examining true positives (TPs) from the gold standard with particularly low log-likelihood scores. The 1,000 TPs with lowest G^2 scores in ENCOW16A were thus also subjected to closer scrutiny. The histogram in Fig. 6.6 shows that their low rank is not an issue of data sparseness: most of the candidates have $f \geq 10$, a substantial portion even $f \geq 100$; but a considerable number of high-frequency pairs occur *less often than expected* in ENCOW16A.

In the list, we find some problematic items, where the gold standard is slightly dubious, e.g. *evidence* + *grow*, which is not impossible but rare compared to the much more common *growing evidence*, where it would be problematic to say that *evidence* is the subject of the verb *grow*.

Many of the low-ranked pairs contain frequent general-purpose verbs (*be*, *go*, *come*, *say*) and relatively frequent nouns (*website*, *problem*, *company*, *system*). Sometimes, skewage in the corpus may be responsible for the low values, for instance, the word *website* occurs roughly 200,000 times with the verb *adhere* and roughly 250,000 times with the verb *use* in the top 1,000 list. This means of course that the expected frequency of the combination *website* + *be* goes up to unnaturally high levels, so that it occurs less frequently than expected (roughly 29,000 hits).

Some of the items are listed with extremely low frequencies, which may be due to parsing/tagging errors. This is particularly obvious in examples such as *tiger* + *spring* or *duck* + *nest*, where the verb was often analysed as a noun by the parsers.

[19] The list for CoreNLP enhanced++ only contains four of them.

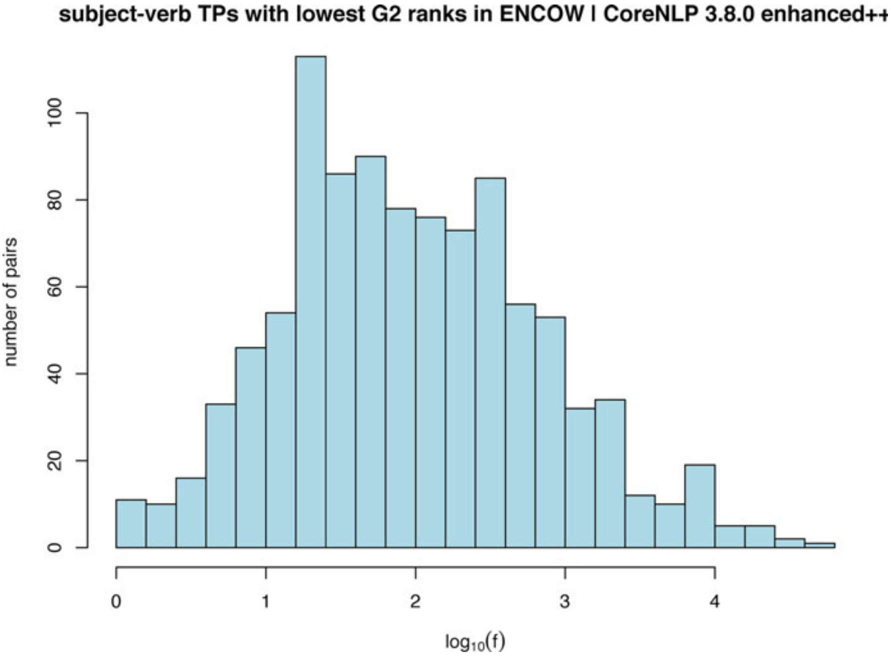

Fig. 6.6 Histogram of the 1,000 lowest-ranked true positive subject-verb collocations on ENCOW16A with CoreNLP enhanced++

5.3 Verb-Object

Examples:

(17) She won the match.

(18) The first *match* was *won* by the Dutch champion.

Overall, the differences between the various parsers are small when it comes to verb-object collocations. The best performance is offered by spaCy and nlp4j, the worst by C&C. Surprisingly, C&C collapsed dependencies are usually slightly worse than the default model used by C&C.

In terms of association measures, we can observe that log-likelihood is slightly better than t-score on the BNC. These differences disappear in ENCOW16A. $\mathrm{MI}_{\mathrm{conf}}$ is substantially worse than log-likelihood and t-score, particularly for short candidate lists; however, $\mathrm{MI}_{\mathrm{conf}}$'s performance improves significantly with the application of a frequency threshold in ENCOW, even though it never reaches the performance of log-likelihood or t-score.

5.4 Adjective-Noun

Examples:

(19) Her boyfriend is really handsome.
(20) He is a very *handsome man*.

Again, the results are very similar for various parsers. Here spaCy wins, but nlp4j does not perform above average, most likely because it does not differentiate between adjectival and nominal modifiers and thus loses precision offered by most other parsers. CoreNLP's results are relatively poor.

On the BNC with t-score, Malt wins for very short candidate lists (up to 10% recall) and is generally quite good (whereas for other relations, it is usually part of the low-performing group).

For ENCOW16A, t-score is slightly better than log-likelihood for very short candidate lists (up to 10% recall). However, t-score takes the biggest hit when dependency relations are not filtered; the other association measures perform only minimally worse. Since spaCy remains the best parser in this condition, we can state that it seems to be excellent both at labelled and unlabelled attachment.

5.5 Verb-Adjective

Examples:

(21) This sounds ingenious.
(22) He pleaded innocent.

Overall, there is very little data for this type of collocation simply because it is comparatively rare. We can observe very high precision, which may indicate that there is only limited variability in both slots. Verb-adjective collocations are the only ones for which simple co-occurrence frequency performs better than any of the association measures. $\mathrm{MI}_{\mathrm{conf}}$'s statistics seem to be particularly bad for this type of construction.

In terms of parsers, C&C and mate-tools win. On ENCOW16A nlp4j performs best for short candidate lists.

5.6 Verb-Adverb

Example:

(23) He brutally assaulted her.

The best-performing parser are spaCy, nlp4j and CoreNLP, but generally there is little difference between the parsers, except for mate and C&C, both of which deliver a recall value of almost 10 percentage points below that of other parsers. For the BNC, the frequency threshold does not make much of a difference, but for ENCOW16A, the image is reversed: Without the frequency threshold, $\mathrm{MI}_{\mathrm{conf}}$ performs worst among the association measures; with a frequency threshold of 50/G, $\mathrm{MI}_{\mathrm{conf}}$ performs best. Log-likelihood outperforms t-score in both conditions.

Interestingly, C&C becomes the best parser (though still with a slightly lower recall than most others) when dependency relations are not filtered, which suggests that the labelled attachment causes trouble here.

5.7 Adverb-Adjective

Example:

(24) He is a *highly capable* manager.

We can observe that Malt is generally bad for this type of collocation. OpenNLP with Stanford Converter, CoreNLP and SyntaxNet are fairly close to one another in their results and usually perform neither particularly well nor particularly badly. The best parsers are spaCy, nlp4j and C&C.

Again, log-likelihood performs best in most conditions and is only outperformed by $\mathrm{MI}_{\mathrm{conf}}$ for short candidate lists with a high frequency threshold of 50/G on ENCOW16A.

6 Conclusion

In this chapter, we have shown that there are no simple solutions for the best possible way to extract collocation candidates. Nonetheless, we can recommend certain practices over others on the basis of our research. Overall, spaCy is a robust parser with good results on all relations. On some specific relations (e.g. subject-verb), it is outperformed by other parsers, but there is no relation where spaCy shows a real weakness. Usually it is part of the leading group in the graph, and it achieves most often the best average precision at 50% recall (AP50).

As for the association measures, we can say that overall log-likelihood is an association measure that works well on all relations even though for some types of collocations, other measures surpass it, e.g. t-score for adjective-noun, $\mathrm{MI}_{\mathrm{conf}}$ for verb-adverb or co-occurrence frequency for verb-adjective. Thus for general-purpose collocation research, we can recommend log-likelihood. For maximum precision for particular relations, for instance, in software used for lexicographic purposes, it would be beneficial to select different association measures for the different relations.

References

Ambati, B. R., Reddy, S., & Kilgarriff, A. (2012). Word sketches for Turkish. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)* (pp. 2945–2950). Istanbul: European Language Resources Association http://www.lrec-conf.org/proceedings/lrec2012/pdf/585_Paper.pdf.

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., & Collins, M. (2016). Globally normalized transition-based neural networks. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (ACL'16)* (pp. 2442–2452). Berlin: Association for Computational Linguistics http://aclweb.org/anthology/P16-1231.

Bartsch, S. (2004). *Structural and functional properties of collocations in English. A corpus study of lexical and pragmatic constraints on lexical co-occurrence*. Tübingen: Narr.

Bartsch, S., & Evert, S. (2014). Towards a Firthian notion of collocation. *OPAL – Online publizierte Arbeiten zur Linguistik, 2*(2014), 48–61 http://pub.ids-mannheim.de/laufend/opal/pdf/opal2014-2.pdf.

Basili, R., Pazienza, M. T., & Velardi, P. (1994). A 'not-so-shallow' parser for collocational analysis. In *Proceedings of the 15th conference on computational linguistics (COLING'94)* (pp. 447–453). Tokyo: Association for Computational Linguistics http://aclweb.org/anthology/C94-1074.

Blaheta, D., & Johnson, M. (2001). Unsupervised learning of multi-word verbs. In *Proceedings of the ACL workshop on collocation: Computational extraction, analysis and exploitation* (pp. 54–60). Toulouse.: http://web.science.mq.edu.au/∼mjohnson/papers/2001/dpb-colloc01.pdf.

Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP'14)* (pp. 740–750). Doha: Association for Computational Linguistics http://aclweb.org/anthology/D14-1082.

Choi, J. D., & McCallum, A. (2013). Transition-based dependency parsing with Selectional branching. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics (ACL'13)* (pp. 1052–1062). Sofia: Association for Computational Linguistics http://aclweb.org/anthology/P13-1104.

Choi, J. D., & Palmer, M. (2011). Getting the most out of transition-based dependency parsing. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human language technologies (ACL'11)* (pp. 687–692). Portland: Association for Computational Linguistics http://aclweb.org/anthology/P11-2121.

Choi, J. D., & Palmer, M. (2012). *Guidelines for the CLEARStyle Constituent to Dependency Conversion*. Institute of Cognitive Science Technical Report 01-12, University of Colorado Boulder.

Church, K., Gale, W., Hanks, P., & Hindle, D. (1989). Parsing, word associations and typical predicate-argument relations. In *Speech and natural language: Proceedings of a workshop held at cape cod, Massachusetts, October 15-18, 1989* (pp. 75–81). Cape Cod.: http://aclweb.org/anthology/H89-2012.

Clark, S., & Curran, J. R. (2007). Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics, 33*(4), 493–556 http://aclweb.org/anthology/J07-4004.

Evert, S. (2004). *The statistics of word Cooccurrences. Word pairs and collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart. Published in 2005 http://elib.uni-stuttgart.de/opus/volltexte/2005/2371/.

Evert, S., & Krenn, B. (2001). Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics (ACL'01)* (pp. 188–195). Toulouse: Association for Computational Linguistics http://www.aclweb.org/anthology/P01-1025.

Evert, S., Uhrig, P., Bartsch, S., & Proisl, T. (2017). E-VIEW-alation – A large-scale evaluation study of association measures for collocation identification. In *Proceedings of eLex 2017 – Electronic lexicography in the 21st century: Lexicography from Scratch* (pp. 531–549). Leiden: Lexical Computing https://elex.link/elex2017/wp-content/uploads/2017/09/paper32.pdf.

Farahmand, M., & Henderson, J. (2016). Modeling the non-substitutability of multiword expressions with distributional semantics and a log-linear model. In *Proceedings of the 12th workshop on multiword expressions* (pp. 61–66). Berlin: Association for Computational Linguistics https://aclweb.org/anthology/W16-1809.

Gries, S. T. (2013). 50-something years of work on collocations: What is or should be next .... *International Journal of Corpus Linguistics, 18*(1), 137–165.

Gries, S. T., & Stefanowitsch, A. (2004). Covarying collexemes in the into-causative. In M. Achard & S. Kemmer (Eds.), *Language, culture, and mind* (pp. 225–236). Stanford, CA: CSLI.

Heid, U., Fritzinger, F., Hauptmann, S., Weidenkaff, J., Weller, M. (2008). Providing corpus data for a dictionary for German juridical phraseology. In Storrer, A., Geyken, A., Siebert, A., Würzner, K-M, Text resources and lexical knowledge. Selected papers from the 9th conference on natural language processing, KONVENS 2008, Berlin, Germany (pp. 131–144). Berlin/Boston: Mouton de Gruyter. https://doi.org/10.1515/9783110211818.2.131

Herbst, T. (1996). What are collocations: Sandy beaches or false teeth? In *English studies* (Vol. 1996/4, pp. 379–393).

Ivanova, K., Heid, U., Walde, S. S. i., Kilgarriff, A., & Pomikalek, J. (2008). Evaluating a German sketch grammar: A case study on noun phrase case. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, & D. Tapias (Eds.), *Proceedings of the sixth international conference on language resources and evaluation (LREC'08)*. Marrakech: European Language Resources Association, 2101–2107 http://www.lrec-conf.org/proceedings/lrec2008/pdf/537_paper.pdf.

Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007* (pp. 105–112). Tartu.: http://dspace.ut.ee/bitstream/handle/10062/2560/reg-Johansson-10.pdf.

Johnson, M. (1999). *Confidence intervals on likelihood estimates for estimating association strengths*. Unpublished technical report.

Katz, G., & Giesbrecht, E. (2006). 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 (MWE'06)* (pp. 12–19). Sydney: Association for Computational Linguistics http://aclweb.org/anthology/W06-1203.

Kermes, H., & Heid, U. (2003). Using chunked corpora for the acquisition of collocations and idiomatic expressions. In F. Kiefer & J. Pajzs (Eds.), *Proceedings of 7th conference on computational lexicography and Corpus research*. Budapest: Research Institute for Linguistics, Hungarian Academy of Sciences.

Kiela, D., & Clark, S. (2013). Detecting compositionality of multi-word expressions using nearest neighbours in vector space models. In *Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP'13)* (pp. 1427–1432). Seattle: Association for Computational Linguistics http://www.aclweb.org/anthology/D13-1147.

Kilgarriff, A., Rychlý, P., Smrz, P., & Tugwell, D. (2004). The sketch engine. In G. Williams & S. Vessier (Eds.), *Proceedings of the 11th EURALEX international congress* (pp. 105–115). Lorient: Université de Bretagne-Sud, Faculté des lettres et des sciences humaines http://euralex.org/wp-content/themes/euralex/proceedings/Euralex%202004/011_2004_V1_Adam%20KILGARRIFF,%20Pavel%20RYCHLY,%20Pavel%20SMRZ,%20David%20TUGWELL_The%20%20Sketch%20Engine.pdf.

Kilgarriff, A., Rychlý, P., Jakubicek, M., Kovář, V., Baisa, V., & Kocincová, L. (2014). Extrinsic corpus evaluation with a collocation dictionary task. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)*. Reykjavik: European Language Resources Association http://www.lrec-conf.org/proceedings/lrec2014/pdf/52_Paper.pdf.

Klotz, M., & Herbst, T. (2016). *English dictionaries: A linguistic introduction*. Berlin: Erich Schmidt.

Lin, D. (1998). Extracting collocations from text corpora. In *Proceedings of the first workshop on computational terminology* (pp. 57–63). Montreal.

Lin, D. (1999). Automatic identification of non-compositional phrases. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL'99)* (pp. 317–324). Morristown: Association for Computational Linguistics http://aclweb.org/anthology/P99-1041.

Lü, Y., & Zhou, M. (2004). Collocation translation acquisition using monolingual corpora. In *Proceedings of the 42nd meeting of the Association for Computational Linguistics (ACL'04)* (pp. 167–174). Barcelona: Association for Computational Linguistics http://aclweb.org/anthology/P04-1022.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics (ACL'14)* (pp. 55–60). Baltimore: Association for Computational Linguistics http://aclweb.org/anthology/P14-5010.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics, 19*(2), 313–330 http://aclweb.org/anthology/J93-2004.

Marneffe, M.-C. de & Manning, C. D. (2008). Stanford dependencies manual. https://nlp.stanford.edu/software/dependencies_manual.pd

Nerima, L., Seretan, V., & Wehrli, E. (2003). Creating a multilingual collocations dictionary from large text corpora. In *Companion volume to the proceedings of the 10th conference of the European chapter of the Association for Computational Linguistics (EACL'03)* (pp. 131–134). Budapest: Association for Computational Linguistics http://aclweb.org/anthology/E03-1022.

Nissim, Malvina, Andrea Zaninello (2013): "Modeling the internal variability of multi-word expressions through a pattern-based method." *ACM Transactions on Speech and Language Processing (TSLP)* 10/2: 7:1–7:26. https://doi.org/10.1145/2483691.2483696

Nivre, J. (2009). Non-projective dependency parsing in expected linear time. In *Proceedings of the 47th annual meeting of the Association for Computational Linguistics and the 4th international joint conference on natural language processing of the AFNLP (ACL'09)* (pp. 351–359). Singapore: Association for Computational Linguistics http://www.aclweb.org/anthology/P09-1040.

Pearce, D. (2001). Synonymy in collocation extraction. In *Proceedings of the NAACL workshop on WordNet and other lexical resources: Applications, extensions and customizations* (pp. 41–46). Pittsburgh: Association for Computational Linguistics.

Pearce, D. (2002). A comparative evaluation of collocation extraction techniques. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02). Las Palmas: European language resources association* (pp. 1530–1536). http://www.lrec-conf.org/proceedings/lrec2002/pdf/169.pdf.

Pecina, P. (2005). An extensive empirical study of collocation extraction methods. In *Proceedings of the ACL student research workshop* (pp. 13–18). Ann Arbor: Association for Computational Linguistic http://aclweb.org/anthology/P05-2003.

Pecina, P. (2010). Lexical association measures and collocation extraction. *Language Resources and Evaluation, 44*, 137–158 https://doi.org/10.1007/s10579-009-9101-4.

Pecina, P., & Schlesinger, P. (2006). Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL 2006 main conference poster sessions* (pp. 651–658). Sydney: Association for Computational Linguistics http://aclweb.org/anthology/P06-2084.

Rodríguez-Fernández, S., Anke, L. E., Carlini, R., & Wanner, L. (2016). Semantics-driven recognition of collocations using word embeddings. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 499–505). Berlin: Association for Computational Linguistics https://doi.org/10.18653/v1/P16-2081.

Sangati, F., & van Cranenburgh, A. (2015). Multiword expression identification with recurring tree fragments and association measures. In *Proceedings of the 11th workshop on multiword expressions* (pp. 10–18). Denver: Association for Computational Linguistics https://doi.org/10.3115/v1/W15-0902.

Schäfer, R. (2015). Processing and querying large web corpora with the COW14 architecture. In P. Bański, H. Biber, E. Breiteneder, M. Kupietz, H. Lüngen, & A. Witt (Eds.), *Proceedings of the 3rd workshop on challenges in the Management of Large Corpora (CMLC-3)* (pp. 28–34). Mannheim: IDS Publication Server https://ids-pub.bsz-bw.de/files/3826/Schaefer_Processing_and_querying_large_web_corpora_2015.pdf.

Schäfer, R., & Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)* (pp. 486–493). Istanbul: European Language Resources Association http://www.lrec-conf.org/proceedings/lrec2012/pdf/834_Paper.pdf.

Schulte im Walde, S. (2003). A collocation database for German verbs and nouns. In *Proceedings of the 7th conference on computational lexicography and text research (COMPLEX'03)* (pp. 73–81). Budapest.: http://www.ims.uni-stuttgart.de/institut/mitarbeiter/schulte/publications/workshop/complex-03.pdf.

Schuster, S., & Manning, C. D. (2016). Enhanced English universal dependencies: An improved representation for natural language understanding tasks. In N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (pp. 2371–2378). Portorož: European Language Resources Association http://www.lrec-conf.org/proceedings/lrec2016/pdf/779_Paper.pdf.

Seretan, V. (2008). *Collocation extraction based on syntactic parsing*. Ph.D. thesis, Faculté des lettres, Université de Genève http://www.issco.unige.ch/en/staff/seretan/publ/PhDThesis-VioletaSeretan.pdf.

Seretan, V., & Wehrli, E. (2006). Accurate collocation extraction using a multilingual parser. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics* (pp. 953–960). Sydney: Association for Computational Linguistics http://aclweb.org/anthology/P06-1120.

Seretan, V., Nerima, L., & Wehrli, E. (2003). Extraction of multi-word collocations using syntactic bigram composition. In *Proceedings of the fourth international conference on recent advances in NLP (RANLP-2003)* (pp. 424–431). https://archive-ouverte.unige.ch/unige:17034.

Seretan, V., Nerima, L., & Wehrli, E. (2004). Multi-word collocation extraction by syntactic composition of collocation bigrams. In N. Nicolov, K. Bontcheva, G. Angelova, & R. Mitkov (Eds.), *Recent advances in natural language processing III. Selected papers from RANLP 2003* (pp. 91–100). Amsterdam/Philadelphia: John Benjamins https://doi.org/10.1075/cilt.260.10ser.

Squillante, L. (2014). Towards an empirical subcategorization of multiword expressions. In *Proceedings of the 10th workshop on multiword expressions (MWE 2014)* (pp. 77–81). Gothenburg: Association for Computational Linguistics http://www.aclweb.org/anthology/W14-0813.

Steedman, M. (2000). *The syntactic process*. Cambridge, MA: The MIT Press.

Stefanowitsch, A., & Gries, S. T. (2005). Covarying collexemes. *Corpus Linguistics and Linguistic Theory, 1*(1), 1–43. https://doi.org/10.1515/cllt.2005.1.1.1.

Stefanowitsch, A., & Gries, S. T. (2009). Corpora and grammar. In A. Lüdeling & M. Kytö (Eds.), *Corpus linguistics: An international handbook* (Vol. 2, pp. 933–952). Berlin, DE/New York, NY: Walter de Gruyter.

Teufel, S., & Grefenstette, G. (1995). Corpus-based method for automatic identification of support verbs for nominalizations. In *Proceedings of the seventh conference of the European chapter of the Association for Computational Linguistics (EACL'95)* (pp. 98–103). Dublin: Association for Computational Linguistics http://aclweb.org/anthology/E95-1014.

Tsvetkov, Y., & Wintner, S. (2014). Identification of multiword expressions by combining multiple linguistic information sources. *Computational Linguistics, 40*(2), 449–468 https://doi.org/10.1162/COLI_a_00177.

Uhrig, P., & Proisl, T. (2012). Less hay, more needles – Using dependency-annotated corpora to provide lexicographers with more accurate lists of collocation candidates. *Lexicographica, 28*, 141–180 https://doi.org/10.1515/lexi.2012-0009.

Villada, M., & Begoña, M. (2005). *Data-driven identification of fixed expressions and their modifiability*. Ph.D. thesis, University of Groningen http://www.rug.nl/research/portal/files/9790774/thesis.pdf.

Weller, M., & Heid, U. (2010). Extraction of German multiword expressions from parsed corpora using context features. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, & D. Tapias (Eds.), *Proceedings of the seventh international conference on language resources and evaluation (LREC'10)* (pp. 3195–3201). Valletta: European Language Resources Association http://lrec-conf.org/proceedings/lrec2010/pdf/428_Paper.pdf.

Wermter, J., & Hahn, U. (2006). You can't beat frequency (unless you use linguistic knowledge) – A qualitative evaluation of association measures for collocation and term extraction. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the ACL (ACL'06)* (pp. 785–792). Sydney: Association for Computational Linguistics http://aclweb.org/anthology/P06-1099.

Wiechmann, D. (2008). On the computation of collostruction strength: Testing measures of association as expressions of lexical bias. *Corpus Linguistics and Linguistic Theory, 4*(2), 253–290 https://doi.org/10.1515/CLLT.2008.011.

Yazdani, M., Farahmand, M., & Henderson, J. (2015). Learning semantic composition to detect non-compositionality of multiword expressions. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP'15)* (pp. 1733–1742). Lisbon: Association for Computational Linguistics http://www.aclweb.org/anthology/D15-1201.

Zinsmeister, H., & Heid, U. (2003). Significant triples: Adjective+noun+verb combinations. In *Proceedings of the 7th conference on computational lexicography and text research (complex 2003)*. Budapest.: http://www.ims.uni-stuttgart.de/%7Ezinsmeis/pubs/SigColl-paper.pdf.

Zinsmeister, H., & Heid, U. (2004). Collocations of complex nouns: Evidence for lexicalisation. In *Proceedings of KONVENS 2004*. Vienna.: https://pdfs.semanticscholar.org/3e5d/d62cbe41b8aa4bbdf37231b85b9b7ef7d94e.pdf.

Dictionaries

OALD8 = *Oxford Advanced Learner's Dictionary of Current English*, 8th edition (2010). Edited by Joanna Turnbull. Oxford: Oxford University Press.

OCD2 = *Oxford Collocations Dictionary for Students of English*, 2nd edition (2009). Edited by Colin MacIntosh. Oxford: Oxford University Press.