Computational phraseology discovery in corpora with the mwetoolkit

Carlos Ramisch

Abstract

Computer tools can help discovering new phraseological units in corpora, thanks to their ability to quickly draw statistics from large amounts of textual data. While the research community has focused on developing and evaluating original algorithms for the automatic discovery of phraseological units, little has been done to transform these sophisticated methods into usable software. In this chapter, we present a brief survey of the main approaches to computational phraseology available. Furthermore, we provide worked out examples of how to apply these methods using the `mwetoolkit`, a free software for the discovery and identification of multiword expressions. The usefulness of the automatically extracted units depends on various factors such as language, corpus size, target units, and available taggers and parsers. Nonetheless, the mwetoolkit allows fine-grained tuning so that this variability is taken into account, adapting the tool to the specificities of each lexicographic environment.

Résumé

Les outils informatiques peuvent assister la découverte de nouvelles unités phraséologiques dans les corpus grâce à leur facilité pour calculer rapidement des statistiques à partir de grands volumes de données textuelles. Alors que la communauté de recherche s'est concentrée sur le développement et l'évaluation d'algorithmes originaux pour la découverte automatique d'unités phraséologiques, la transformation de ces méthodes sophistiquées en logiciels utilisables est souvent ignorée. Ce chapitre présente un bref résumé des principales approches informatiques disponibles pour la découverte d'unités phraséologiques. Nous présenterons des exemples détaillés de l'application de ces approches avec le `mwetoolkit`, un logiciel libre pour la découverte et l'identification d'unités polylexicales. L'utilité des unités extraites automatiquement dépend de plusieurs facteurs comme la langue, la taille du corpus, les unités cibles, et les étiqueteurs et analyseurs disponibles. Néanmoins, le mwetoolkit permet un paramétrage fin, de manière à ce que cette variabilité soit prise en compte dans l'adaptation de l'outil à chaque environnement lexicographique.

Keywords: *phraseological units, automatic phraseology discovery, morphosyntactic patterns, association scores, mwetoolkit*

1 Introduction

Phraseological units are pervasive in human languages. They range from compound nominals (*prime minister*) to complex formulaic templates (*looking forward to hearing from you*), including idiomatic expressions (*to make ends meet*) and collocations (*heavy rain*). While easily mastered by native speakers, they pose challenges for foreign language learners, as their use confers naturalness to discourse, even though they are often unpredictable. Particularly in specialised domains, phraseological units are numerous and employing them appropriately is crucial in technical and scientific communication. For all these reasons, compiling phraseology dictionaries is an absolute need, to account for the pervasiveness of multiword phenomena in languages.

Phraseology is crucial not only for lexicography, but also for computational linguistics. Indeed, the development of natural language processing (NLP) applications often relies on appropriately representing and processing phraseological units in machine-readable lexicons and grammars (Sag et al., 2002). In the NLP community, there has been much enthusiasm about so-called *multiword expressions*. Since the beginning of the 2000's, numerous algorithms and experiments have been published for the automatic processing of multiword units, covering their automatic discovery, in-context identification, syntactic and semantic analysis, as well as translation (Markantonatou et al., 2017). Synergies between NLP and lexicography are a natural consequence of this mutual interest in phraseology.

However, given the different backgrounds, goals and traditions of research communities, cooperation presents some challenges, for example due to the lack of a homogenized terminology. One question regards the overlaps among phraseological units, collocations and multiword expressions. While these terms are undoubtedly related, it is not straightforward to clearly delineate the subtle differences in the phenomena they cover.

While we will not address the issue of competing definitions, we will instead consider that phraseological units are groups of lexemes that present some idiosyncrasy with respect to ordinary word combinations (Baldwin and Kim, 2010), so that their particularities must be recorded in a lexicon (Evert, 2004). Phraseology lexicons, in turn, are useful both for humans and computers, for robust and fluent analysis and generation of language.

The manual construction of lexical resources that include phraseological units is often onerous and time-consuming. It requires not only lexicographic expertise but also corpus-based work, because many units have non-standard properties that only emerge from the study of their use in context. Computers are often employed to enhance, speed up and generally assist in lexicographic tasks, given that they can quickly process large amounts of text (Dagan and Church, 1994; Heid, 2008). Thus, computational systems play a double role in the creation of phraseological resources. On the one hand, they *use* these resources in NLP tasks and applications such as parsing, machine translation and information extraction. On the other hand, computational systems also *support* the creation of lexical resources. This chapter focuses on the latter interaction, when computers are used to help building phraseological lexicons, which we will henceforth refer to as **phraseology discovery**.[1]

In computational linguistics, much has been said about the automatic discovery of phraseological units in corpora using computational tools (Evert and Krenn, 2005; Seretan, 2011; Ramisch et al., 2013; Ramisch, 2015). In spite of a huge literature, newcomers to the field often feel frustrated about the use of computational tools for phraseology discovery. To date, there is no simple and direct answer to the question *What are the best freely available computational tools to help building phraseological resources?*

While solutions do exist, tools for computer-assisted phraseology are often not freely available. When they are, many are hard to use, limited to a few languages and processing systems, and do not always implement more sophisticated techniques reported in research papers. To make things eve more complicated, when available tools exist, they often depend on a given syntactic formalism, data format, language-specific configurations and are not adaptable or portable to different scenarios (language, domain, type of target phraseological unit). Users often have to choose between powerful but extremely complex systems that require computational expertise, and systems that are easy to use but no not allow fine-grained customisation and do not always implement the latest research advances.

This chapter provides a brief survey of current techniques for the automatic discovery of phraseology in monolingual text bases from a computational perspective. We will pay special attention to the availability

[1] Many terms have been used to denote the task of finding new phraseological units in text, including *phraseology discovery, identification, extraction,* and *acquisition.* We consistently employ *phraseology discovery* even when references might use a different terminology.

of these techniques and their implementation in free software. Each subsection includes worked out examples of candidate phraseological units obtained automatically using the mwetoolkit[2].

After this introductory section, we provide a brief overview of related work in computational phraseology, focusing on tools (§ 2). Then, the main contribution of this work is to discuss a specific tool for multiword phraseology discovery in corpora: the mwetoolkit (§ 3). We present worked out examples of use of the mwetoolkit for phraseology discovery, including candidate search patterns, association scores and other types of scores (§ 4). We close this chapter with a discussion of what are, in our opinion, the main bottlenecks that prevent the techniques described here to be employed in the large-scale production of lexical and phraseological language resources (§ 5).

2 Computational phraseology discovery

Although much has been published about the discovery of multiword units in corpora, not all methods and algorithms yield the publication of corresponding software. Therefore, rather than providing a comprehensive overview of phraseology discovery, this section provides an overview of the general architecture, followed by a summary of methods that have been implemented and released, and are thus directly applicable. Tools such as AMALGrAM (Schneider et al., 2014), LGTagger (Constant and Tellier, 2012), and jMWE (Finlayson and Kulkarni, 2011), for the identification of phraseological units in running text, to which a pre-compiled lexicon is usually provided, are not in the scope of this chapter.

For a more complete survey on phraseology discovery, the different proposed methods and their performances, we refer to Evert (2004); Pecina (2008); Manning and Schütze (1999); McKeown and Radev (1999); Baldwin and Kim (2010); Seretan (2011); Ramisch (2015). In addition to monolingual discovery, other tasks have also been investigated in computational linguistics, such as bilingual phraseology discovery (Ha et al., 2008; Morin and Daille, 2010; Weller and Heid, 2012; Rivera et al., 2013), automatic interpretation and disambiguation of multiword expressions (Fazly et al., 2009) and their integration into applications such as parsing (Constant et al., 2013) and machine translation (Carpuat and Diab, 2010). For further reading, we recommend the proceedings of the annual workshop on multiword expressions (Markantonatou et al., 2017),[3] as well as journal special issues on the topic (Villavicencio et al., 2005; Rayson et al., 2010; Bond et al., 2013; Ramisch et al., 2013).

General architecture Tools for corpus-based phraseology discovery use various strategies and have heterogeneous architectures. Often some **preprocessing** is applied to raw corpora before discovery, minimally by performing spurious content cleaning, sentence splitting, tokenisation, and case homogenisation. Optionally, some tools also employ automatic analysers to enrich the text with part-of-speech tags and, sometimes, automatically generated syntactic trees. The availability of taggers and parsers depends on the target language, so this is not always possible.

After preprocessing, tools extract **candidate** phraseological units from text based on recurring patterns. These patterns may be as simple as n-grams, that is, sequences of n contiguous tokens in a sentence (Pedersen et al., 2011; Silva and Lopes, 1999). Tools can also employ more sophisticated morphosyntactic patterns, such as sequences formed by a noun followed by a preposition and another noun (Kilgarriff et al., 2014). When available, syntactic information can also be used to extract candidates, for instance, focusing on verb-object pair (Martens and Vandeghinste, 2010; Sangati et al., 2010). The choice may depend on the nature of the target phraseological units, and a mixture of these strategies can be preferable

[2] http://mwetoolkit.sf.net

[3] http://multiword.sf.net

(Ramisch, 2015).

After candidate extraction, most available tools offer the possibility to **filter** the lists of candidates by using numerical scores. The idea of filtering is that true phraseological units can be distinguished from false positives by their statistical patterns. The most common filtering strategy is the use of association scores such as the candidate frequency, point-wise mutual information (Church and Hanks, 1990), Student's t score or log-likelihood ratio (Dunning, 1993). Scores are used to rank the candidates, assuming that those with higher scores are more likely to be kept for inclusion in a phraseological lexicon. As we will exemplify later (§ 4.2), other scores, based on contextual and contrastive information, can also help retrieving interesting units.

Freely available tools Tools for phraseology discovery generally mix linguistic analysis and statistical information as clues for finding new units in texts. Here, we present a list of freely available tools that can be used mostly for monolingual MWE acquisition. While most of them require some familiarity with the textual command line interfaces, some also provide a graphical user interface or a web application.

The N-gram Statistics Package (NSP)[4] is a command-line tool for the statistical analysis of n-grams in text files (Pedersen et al., 2011; Banerjee and Pedersen, 2003). It provides scripts for counting n-grams and calculating association scores, where an n-gram is either a sequence of n contiguous words or n words occurring in a window of $w \geq n$ words in a sentence. While most of the measures are only applicable to 2-grams, some of them are also extended to 3-grams and 4-grams, notably the log-likelihood measure. The tool takes as input a raw text corpus and a parameter value fixing the size of the target n-grams, and provides as output a list of candidate units extracted from the corpus along with their counts, which can further be used to calculate association scores.

Analogously, LocalMaxs[5] is a script that extracts candidate units by generating all possible n-grams from a sentence. It further filters them based on the local maxima of a customisable association score distribution (Silva and Lopes, 1999), thus taking into account larger units that contain nested smaller ones. The tool includes a strict version, which prioritises high precision, and a relaxed version, which focuses on high recall. Both NSP and LocalMaxs are based purely on token counts and are completely language independent. On the other hand, there is no direct support to linguistic information such as keeping only n-grams that involve nouns.

Focusing on the retrieval of discontiguous units, Xtract is an algorithm that uses a sliding window of length w to scan the text (Smadja, 1993). It requires the input text to be POS-tagged, so that filters can be applied on the types of extracted candidates. Xtract first generates bigrams by calculating the average distance between words in the sliding window, as well as their standard deviation. Words that tend to occur always in the same position with respect to each other (small standard deviation) are considered as candidates, which are then expanded to larger n-grams. The Dragon toolkit[6] is a Java library that implements the Xtract algorithm and can be included in computational tools (Zhou et al., 2007).

The `UCS` toolkit[7] is a command-line package to calculate association scores (Evert, 2004). Additionally, it provides powerful mathematical tools like dispersion tests, frequency distribution models and evaluation metrics. UCS focuses on high accuracy calculations for 2-grams, but, unlike the other approaches, it does not extract candidate units from corpora. Instead, it receives a list of candidates and their respective counts as input, relying on external tools for corpus preprocessing and candidate extraction. Then, it

[4] http://search.cpan.org/dist/Text-NSP

[5] http://hlt.di.fct.unl.pt/luis/multiwords/

[6] http://dragon.ischool.drexel.edu/

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

calculates the measures and ranks the candidates. Another tool that works in a similar way is Druid.[8] It is based on distributional similarity models which estimate to what extent a given candidate could be replaced by a single word, assuming that phraseological units convey more atomic, non-decomposable meanings than regular combinations (Riedl and Biemann, 2015).

In contrast with the above-mentioned tools, there are also some tools that are not based on word sequences, but rather work with syntactic trees. Thus, they require syntactically analysed corpora as input, generally preprocessed by an automatic parser. Such tools are specially well suited for the discovery of flexible units such as verbal idioms, formulaic phrases, and collocations. Examples of such tools include Varro (Martens and Vandeghinste, 2010),[9] DiscoDOP(Sangati et al., 2010)[10] and FipsCo(Seretan and Wehrli, 2009).[11]

The tools surveyed here are mostly developed by researchers in computational linguistics for a project or thesis. Hence, their goal is not to optimise ease of use for users that are not necessarily familiar with command-line computational tools. The idea of the Sketch Engine is to make such tools accessible and friendly by providing an intuitive web interface for corpus-based phraseology discovery.[12] Similarly to other tools, it allows loading corpora, preprocessing them with POS tags and lemmas, and then extracting co-occurrence patterns (the “sketches”) based on morphosyntactic patterns and association scores (Kilgarriff et al., 2014). On the other hand, since it is not free, but commercialised by a company, so special attention is given to the presentation of results, user support and providing useful tools for corpora work (e.g. a tool for crawling web corpora for a given language and domain).

There are also numerous freely available web services and downloadable tools for automatic term extraction, not necessarily focusing on phraseology (Drouin, 2004; Heid et al., 2010). These tools are generally language dependent, having versions for major European languages like English, Spanish, French and Italian. Examples of such tools are TermoStat[13], AntConc[14] and TerMine.[15] Most of them are provide user-friendly graphical interfaces or direct web access. On the other hand, they do not always allow fine tuning of discovery parameters.

3 The mwetoolkit

In spite of the existence of a certain number of available tools for phraseology discovery, they usually only deal with part of the discovery process. For example, while UCS provides several association scores for candidate ranking, the extraction of candidates from the corpus needs to be performed externally. NSP provides support for larger n-grams, but it is impossible to describe more linguistically motivated extraction patterns based on parts of speech, lemmas or syntactic relations (Ramisch et al., 2012).

In a context where existing methods only implemented part of what we needed, we wanted to conceive a *generic methodology* that would cover the whole discovery process. The mwetoolkit is a tool designed to perform automatic discovery of multiword units in specialised and general-purpose corpora (Ramisch et al., 2010b,a; Ramisch, 2015). It implements hybrid knowledge-poor techniques that can be applied virtually to any corpus, independently of the domain and of the language. The goal of the mwetoolkit

[8] http://ltmaggie.informatik.uni-hamburg.de/jobimtext/components/druid/

[9] http://sourceforge.net/projects/varro/

[10] https://github.com/andreasvc/disco-dop

[11] http://129.194.38.128:81/FipsCoView

[12] https://www.sketchengine.co.uk/

[13] http://olst.ling.umontreal.ca/~drouinp/termostat_web/

[14] http://www.antlab.sci.waseda.ac.jp/software.html

[15] http://www.nactem.ac.uk/software/termine/

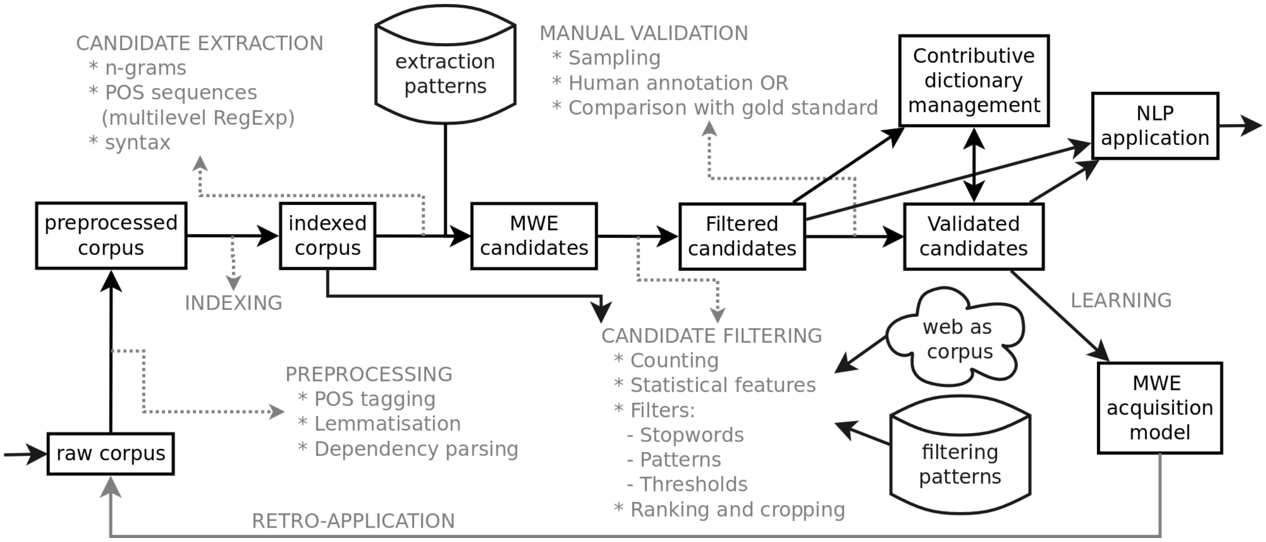


Figure 1: The mwetoolkit modules and their chaining in MWE discovery.

is to aid lexicographers and terminographers in the challenging task of creating language resources that include multiword entries.[16]

Given the plethora of computational discovery methods available and the lack of consensus about their performances, the methodology implemented by the mwetoolkit should necessarily allow *multiple solutions* for a given sub-task. Thus, decisions such as the level of linguistic analysis, length n of the n-grams, filtering thresholds and evaluation measures should not be made by the method itself, but users should be able to chose and tune the parameters according to the needs. This implies that the tool does not provide a push-button method, but one that can be adapted and tuned to a large number of contexts, maximising its *portability*.

The mwetoolkit adopts the standard sub-task definition consisting of two phases: candidate extraction and candidate filtering. In the first phase, one acquires candidates based either on flat n-grams or specific morphosyntactic patterns (of surface forms, lemmas, POS tags and dependency relations). Once the candidate lists are extracted, it is possible to filter them by defining criteria that range from simple count-based thresholds, to more complex features such as association and semantic compositionality scores. Since some scores are based on corpus word and n-gram counts, the toolkit provides both a corpus indexing facility and integration with web search engines (for using the web as a corpus). Additionally, for the evaluation phase, we provide validation and annotation facilities.

The mwetoolkit methodology was implemented as a set of independent modules, each taking the form of a separate Python script, that handle an intermediary representation of the *corpus*, the list of MWE *patterns*, and the list of MWE *candidates*. Each module performs a specific task in the discovery process, from the raw corpus to the filtered list of candidates and their associated scores. Figure 1 summarises the architecture of mwetoolkit, which will be exemplified in section 4. Examples of applications of the mwetoolkit include the discovery of nominal expressions in Greek (Linardaki et al., 2010), of complex predicates in Portuguese (Duran et al., 2011), and of idiomatic nominal compounds in French and English (Cordeiro et al., 2016a).

[16] http://mwetoolkit.sf.net

This is a pre-print of an article published in *Computational Phraseology* by John Benjamins Publishers.
The final authenticated version is available online at: https://doi.org/10.1075/ivitra.24.06ram

4 Phraseology discovery with the mwetoolkit

In the following subsections, we detail existing methods commonly employed in phraseology discovery. Each method will be exemplified using the mwetoolkit. Our goal is to illustrate how the discussed discovery method works rather than presenting original results in phraseology discovery *per se*. Therefore, our experimental set-up is simplistic: we work on a fragment of the English UKWaC that is made freely available including POS tags, lemmas and syntactic dependencies.[17] This fragment (henceforth UKWaC-frg) consists of the first 100,000 sentences of the corpus, which represent around 2.6 million tokens of general-purpose texts crawled from the British world wide web and cleaned automatically (Baroni and Bernardini, 2006).

We underline that the artificial results reported in toy experiments in the remainder of this section were not tuned for optimal performance. Moreover, the mwetoolkit is language independent and can deal with raw corpora in any space-separated writing system, optionally handling any POS tagset, lemmatizer and dependency syntax format that may be applied to the corpus.

4.1 Candidate search patterns

Candidate phraseological units can be extracted from automatically POS-tagged, lemmatized and/or parsed corpora using ad hoc scripts. However, more principled corpus queries are often necessary to obtain lists of candidates to include in phraseological dictionaries. These corpus queries need to deal with multi-level annotations in the corpus, which can be very helpful in phraseology discovery. For instance, Seretan (2011) has shown that syntactic relations can be used in collocation discovery patterns to overcome the limitations of shallow POS sequence patterns.

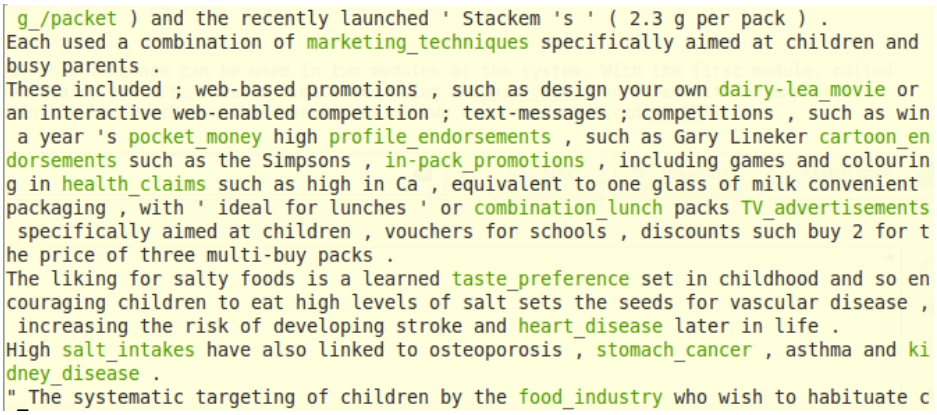
As a consequence, phraseology discovery requires a powerful corpus query language that correctly matches user-defined search patterns with the information available in the corpus (Kilgarriff et al., 2014). The expressive power and generality of the query language is determinant for successfully adapting a candidate extraction method that works well in a given configuration to another language, domain and corpus.

The mwetoolkit uses a multi-level regular expression language to express corpus searches. This language can be used in two modules of the system. With the first module, called `grep.py`, it is possible to explore the corpus, finding sentences that match a given search pattern. This is similar to using a concordancer enriched with an expressive query language. For example, assuming that the corpus is contained in a file named `ukwac-frg.moses`, the command below allows finding and showing all sentences in the corpus that contain sequences of two contiguous common nouns (`NN NN`).

```
mwetoolkit/bin/grep.py -e "NN NN" --to XML ukwac-frg.moses |
mwetoolkit/bin/view.py --to PlainCorpus
```

While finding sequences of contiguous POS tags is useful, it is not always sufficient to model the target expressions. The query language of the mwetoolkit allows the use of complex operators adapted from regular expressions. For instance, suppose we are interested in 2-word complex nominals in English, where the modifier that precedes the noun can be either an adjective or another noun. This can be done by defining a pattern whose first member is an alternative, denoted with a pipe symbol (`|`) between the POS for noun and the one for adjective. We exemplify below the use of this pattern as the input for another module, `candidates.py`, which carries out the extraction of candidate units from corpora, yielding a list of candidates like the one shown in the first column of Table 1:

[17] The UK web as corpus: http://wacky.sslmit.unibo.it/doku.php?id=corpora



```
 g_/packet ) and the recently launched ' Stackem 's ' ( 2.3 g per pack ) .
Each used a combination of marketing_techniques specifically aimed at children and
busy parents .
These included ; web-based promotions , such as design your own dairy-lea_movie or
an interactive web-enabled competition ; text-messages ; competitions , such as win
 a year 's pocket_money high profile_endorsements , such as Gary Lineker cartoon_en
dorsements such as the Simpsons , in-pack_promotions , including games and colourin
g in health_claims such as high in Ca , equivalent to one glass of milk convenient
packaging , with ' ideal for lunches ' or combination_lunch packs TV_advertisements
 specifically aimed at children , vouchers for schools , discounts such buy 2 for t
he price of three multi-buy packs .
The liking for salty foods is a learned taste_preference set in childhood and so en
couraging children to eat high levels of salt sets the seeds for vascular disease ,
 increasing the risk of developing stroke and heart_disease later in life .
High salt_intakes have also linked to osteoporosis , stomach_cancer , asthma and ki
dney_disease .
"_The systematic targeting of children by the food_industry who wish to habituate c
```

Figure 2: Query `NN NN`, matching sequences of two common nouns in UKWaC-frg, visualized in a concordancer-like command line interface.

```
mwetoolkit/bin/candidates.py -e "(NN|JJ) NN" ukwac-frg.moses
```

Other operators are available, for instance, to indicate the repetition of elements. For example, the second column of Table 1 shows the discovered candidates corresponding to a pattern similar to the first one, but in which the first element can be repeated an arbitrary number of times, retrieving units such as *large scale show*.

Sometimes, we are interested in the collocation patterns of a specific lexical unit. For instance, we would like to know which nouns and adjectives can modify the word *show*. The third column of The second column of Table 1 exemplifies this, also showing how it is possible to limit the POS of the word *show* to nominal occurrences only, excluding adjectives and nouns which precede the verb *to show*.

In the third column, we show the results of a pattern in which the head noun was replaced by a regular expression: we are interested here in the co-occurrence pattern of words ending with the suffix *ion*. Finally, the fourth and last column shows a syntactic pattern, in which we look for all nouns that can occur as subject of the verb *to say*.

The syntax of the query language is explained on the documentation of the mwetoolkit, but for the sake of completeness we provide below the exact queries that were used to generate the candidates presented in Table 1:

1. `(JJ|NN) NN`: a sequence of exactly two tokens, where the last one corresponds to a common noun (POS starts with *NN*) and the first one is a disjunction between a common noun (*NN*) or an adjective (*JJ*).[18] This captures 2-word noun phrases consisting of a head noun and a modifier noun or adjective.

2. `(JJ|NN){repeat=+} [pos~/NN.*/i lemma='show']`: a sequence of at least two tokens, where the last one corresponds to the common noun *show*[19] and the first one is a disjunction between a common noun (*NN*) or an adjective (*JJ*), which can be repeated more than once. This captures noun phrases where the head noun is *show*, preceded by a sequence of modifier nouns or adjectives.

[18] This pattern uses a short-cut notation for `([pos∼/JJ.*/i]|[pos∼/NN.*/i]) [pos∼/NN.*/i]`, that is, all-capital strings are internally interpreted as case-insensitive POS prefixes.

[19] Here, we exemplify the explicit notation by which one can constraint different token information levels (POS, lemma, etc.) using equality (`=`) or regular expression approximate matching (∼). The trailing `i` stands for case insensitive.

Pattern 1	Pattern 2	Pattern 3	Pattern 4
more information	TV show	more information	official say
more detail	bike show	further information	spokesman say
last year	large scale show	high education	teacher say
further information	scale show	food production	spokeswoman say
first time	sci-fi show	legal information	official say
wide range	next show	above application	report say
same time	research show	additional information	government say
web site	radio show	late version	man say
local authority	successful show	special collection	mp3gadgetblog say
last week	series show	personal information	cannot say
car park	film show	voluntary organisation	minister say
high quality	few show	southern section	newspaper say
many year	trade show	trade union	signature say
next year	reality show	new version	police say
many people	radio show	other organisation	third say
long time		next generation	manager say
few year		public consultation	text say
view guestbook		good condition	
other hand		industrial action	
hard drive		data protection	

Table 1: Examples of candidates extracted from UKWaC-frg using different types of patterns, shown as headers. Only a sample of the candidates that occur twice or is are shown.

3. `(JJ|NN){repeat=+} [lemma~/.*ion/]`: a sequence of at least two tokens, where the last one is any word ending with the suffix `ion` and the first one is a sequence of common nouns or adjectives which may repeat once or more. This captures mainly noun phrases where the head noun is a nominalisation, preceded by a sequence of modifier nouns or adjectives.

4. `[pos~/NN.*/ syndep=SBJ:v] []{repeat=* ignore=1} [pos~/VV.*/ lemma=say]{id=v}`: a sequence of two tokens, of which the first is a common noun and the second is the main verb *to say*. The pattern is discontiguous, and allows any number of intervening words, that will not be considered as part of the matched expression (`ignore=1`). Additionally, there is a syntactic constraint saying that the first token must be the subject (`syndep=SBJ:v`) of the last word, identified by a unique name `id=v`. This captures possible nouns that can occur as the subject of the verb *to say*.

4.2 Association scores

Collocation is a linguistic phenomenon characterised in statistical terms by *outstanding cooccurrence*. That is, words in collocations have a tendency to co-occur more often than it would be expected by pure chance in a corpus, like if they attracted each other. Collocation is an important property that distinguishes phraseological units from regular word combinations.

Dozens of association scores have been proposed to model outstanding cooccurrence in texts (Pecina, 2011). Association scores estimate the strength of association between the words contained in a candidate phraseological unit. They are based on the co-occurrence count and on the individual word counts of the candidate in a large corpus.

frequency	PMI	t-score	Log-likelihood
young people	cerebral palsy	young people	young people
more information	metabolic acidosis	more information	wide range
more detail	john baker	more detail	last year
last year	amino acid	last year	more detail
further information	scheduled uptime	further information	further information
first time	systemic aciclovir	first time	more information
wide range	fortune teller	wide range	local authority
same time	mifid override	same time	web site
web site	injectable medicine	web site	car park
local authority	holy hierarch	local authority	view guestbook
last week	dew pond	last week	first time
car park	asylum seeker	car park	same time
high quality	rackmount cabinet	high quality	last week
next year	cupc4kes reply	next year	hard drive
many year	stainless steel	many year	high quality
many people	holdem poker	view guestbook	long term
view guestbook	carbon dioxide	many people	email address
long time	scheduled break	long time	distributive justice
few year	non-executive director	few year	mental health
other hand	wrongful pregnancy	other hand	health insurance

Table 2: Examples of top-ranked candidates sorted by frequency, and by association scores, extracted from UKWaC-FRGwith pattern `(JJ|NN) NN` .

Suppose a candidate phraseological unit formed by n words, w_1, $w_2 \dots w_n$. Most association scores employed nowadays take into account the observed co-occurrence count $c(w_1 \dots w_n)$ of the whole candidate. In Table 2, we show the cooccurrence counts of the top-n most frequent candidates extracted by the pattern `(JJ|NN) NN` in UKWaC-FRG.[20] All words are lemmatised to neutralise variants, but this may lead to ungrammatical entries such as *many year*, which actually corresponds to occurrences of *many years*, in plural.

The problem of pure co-occurrence is that frequent word combinations can be a result of pure chance because the involved words are very frequent per se, like *and we*, *it is* and *of the* Manning and Schütze (1999). These are usually function words that are not necessarily interesting for phraseology discovery. Even when we restrict the set of acceptable POS tags, like we did in Table 2 to include only adjectives and nouns, very frequent words still tend to diminish the usefulness of the extracted list. For example, modifiers such as *many* and *last* and *more* are combined with frequent words like *information* and *year*, but these are regular combinations with limited phraseological interest.

The problem with co-occurrence frequency is that association scores should also consider the expected count of a word combination $E(w_1 \dots w_n)$, comparing it with the simple co-occurrence count $c(w_1 \dots w_n)$. If the appearances of words w_i in a corpus are modelled as independent events, we expect that the co-

[20] This list is similar to the examples shown in Table 1, but sorted by descending co-occurrence frequency.

occurrence count of a group of words equals the product of their individual probabilities $\frac{c(w_i)}{N}$ scaled by the total corpus size N. Therefore, the expected count E is estimated considering the number of occurrences of individual words in the candidate $c(w_1)$ through $c(w_n)$:

$$E(w_1 \dots w_n) = \frac{c(w_1) \times \dots \times c(w_n)}{N^{n-1}}$$

Pointwise mutual information (PMI) is one of the most popular association scores using this principle. It is not only used for phraseology discovery, but also for many other tasks in computational linguistics. PMI was first proposed in multiword terminology discovery by Church and Hanks (1990). It is the log-ratio between observed and expected counts :[21]

$$PMI(w_1 \dots w_n) = \log \frac{c(w_1 \dots w_n)}{E(w_1 \dots w_n)}.$$

PMI indicates how well the presence of an individual word predicts the presence of the whole phrase, or, in other words, it quantifies the dependence between words. Values close to zero indicate independence and the candidate words are discarded, whereas large values indicate outstanding cooccurrence. The second column of table 2 shows the top-ranked candidates extracted from UKWaC-frg and sorted in descending order of PMI.

If things were simple, then PMI would solve all our problems. However, corpora are often not large enough to cover the phenomenon under study and many interesting combinations are infrequent. Data sparseness is a problem for PMI, since this score tends to over-estimate the importance of rare word combinations formed up by rare words. In Table 2, we show only candidates that co-occur more than 10 times in the toy corpus, to avoid retrieving spurious rare combinations such as foreign words and typos. Even with this restriction, PMI still retrieves quite rare candidates, some of which are actually quite interesting, such as *cerebral palsy*, *amino acid*, *stainless steel* and *carbon dioxide*. Some entries, however, are simply rare words that always appear combined with the same other words, such as *cup4kes reply*, which is probably a text appearing on a website of a cupcakes seller. Variants of PMI were proposed in the literature, trying to increase the importance of the observed count, in order to avoid this kind of problem (Daille, 1995; Riedl and Biemann, 2013).

Some association scores are based on hypothesis testing. Again, assuming independence between words, we can hypothesise that in regular (non-phraseological) word combinations, the observed and expected counts should be identical, that is $H_0 : c(w_1 \dots w_n) = E(w_1 \dots w_n)$. Using a test statistic like Student's t, large values are strong evidence to reject H_0. The third column of Table 2 shows how the t-score ranks the candidate nominal compound candidates extracted from our toy corpus.

More sophisticated tests for two-word MWE candidates take into account their *contingency table*. Examples of such measures are χ^2 and the more robust likelihood ratio (Dunning, 1993). The latter is only applicable to two-word expressions, but usually provides high-quality candidates. An example of ranking by log-likelihood is shown in the fourth and last column of Table 2.

Many other association scores were proposed in the literature, and there is no "silver bullet". While most of them are useful, deciding on which score to use for a given corpus is a matter of trial and error. Therefore, the mwetoolkit calculates several scores using the module `feat_association.py`. Afterwards, users can try different sort orders and decide on one (or several) scores to use in order to filter the candidate entries before starting encoding them in lexicons. It also implements other scores that are briefly surveyed below.

[21] Notice that, since this is the logarithm of a quotient, it is equivalent to $\log c(w_1 \dots w_n) - \log E(w_1 \dots w_n)$. In other words, observed and expected counts are compared through direct subtraction, but in logarithm domain.

4.3 Other scores

While association scores are the mainstream in phraseology discovery, they provide limited information about the behaviour of the target phraseological units, specially when those are not frequent in the available corpora. Other types of scores have been proposed to capture other properties of phraseological units, such as their tendency to appear in specialised texts, their limited variability, their semantic idiomaticity and their limited word-for-word translatability. Here, we provide an overview of these scores, which are also implemented and available in the mwetoolkit. However, since they require complementary resources, we do not show examples but rather provide pointers to publications in which these scores are described and evaluated in more depth.

Contrastive scores are a useful source of complementary information for terminological or, more generally, domain-specific units. Several scores have been proposed to take into account the different frequency distributions of words across domain and general-purpose (contrastive) corpora (Drouin, 2004; Bonin et al., 2010). The intuition here is that units that appear frequently in a specialised corpus but are not common in general-purpose texts are likely to be specialised terms, and this is applicable not only to single words but also to phraseological units.

A simple contrastive score consists in the ratio between the frequency of the candidate in the domain corpus and its frequency in the contrastive corpus. The higher this ratio, the more specialised the candidate is. Since contrastive corpora should be large, they can be replaced by web hit counts, that is, the number of web pages containing a given candidate phrase (Ramisch et al., 2010c). In the mwetoolkit, these scores are implemented in a module named `feat_contrast.py`.

Another class of useful methods are those which try to predict the **variability scores** of candidates. Indeed, one of the properties of phraseological units is that they do not allow full morphological, syntactic, and semantic flexibility as compared to similar free word combinations. Variability scores are a relatively under-exploited method based on automatic variant generation and subsequent (web) corpus searches. In other words, one first generate artificial variants for a given candidate, and then verifies whether these variants are attested in a large corpus or in the web. If variants appear frequently, then it is less likely that the candidate is a frozen phraseological unit. The skewness of the variant distribution can be measured by its entropy: the higher the entropy, the more uniform the distribution is, thus a low entropy score is a hint for more fixed phraseological units. In the mwetoolkit, these scores are implemented in a module named `feat_entropy.py`.

The generation of variants is language specific, and can be performed in several ways. Probably the simplest type of variability score is *permutation entropy*, in which candidates are randomly reordered and then looked up in the web (Villavicencio et al., 2007). A slightly more sophisticated version of this score uses the syntactic behaviour of the expressions in order to create linguistically-informed permutations (Ramisch et al., 2008). Another possibility is to introduce explicit paraphrases, for example, by replacing sequences of the type *noun1 noun2* by a corresponding structure *noun2 PREP noun1*, trying different types of prepositions that correspond to the semantics of the compound (Nakov and Hearst, 2005). Finally, it is also possible to use general-purpose synonym dictionaries such as WordNet to try replacing words in the candidate by synonyms, since fixed phraseological units will generally not accept as much replacement as a regular combination would (Pearce, 2001; Duran and Ramisch, 2011; Duran et al., 2013).

With the growing importance of distributional semantics and word embeddings (Mikolov et al., 2013), new methods have been developed to identify non-compositional combinations in texts (Salehi et al., 2015; Cordeiro et al., 2016b). The basic ideas of *compositionality scores* is to measure the similarity between the candidate phraseological unit and the words that compose it. Therefore, we use distributional models to build vectorial representations for each word and phrase from raw corpora. Afterwards, we use a vector operation (such as vector addition) to create a combined vector representing the sum of the meanings of the component words. Finally, we calculate its similarity (e.g. using the cosine of the angle between the vectors) to estimate how close the vector of the whole candidate unit is to the combined vector of the component words. Compositionality scores have been successfully employed to discover non-compositional phrases such as idiomatic noun compounds in English, French (Cordeiro et al., 2016a) and German (Roller et al., 2013), and verbal expressions in English (Cook and Stevenson, 2010) and in German (Köper et al., 2016). In the mwetoolkit, these scores are implemented in a module named `feat_compositionality.py` (Cordeiro et al., 2016b).

Phraseological units often cannot be translated word by word. Therefore, parallel corpora can be a useful source of information for phraseological units. For example, one can use automatic word alignment tools to locate phrases in which the alignment is not perfect, indicating the presence of a phraseological unit (de Medeiros Caseli et al., 2009; Tsvetkov and Wintner, 2011; Salehi and Cook, 2013). Additionally, it is possible to use a method similar to the one for variant generation for translation generation: if a candidate cannot be translated word-by-word, then it is probably a phraseological unit (Morin et al., 2007; Vargas et al., 2017). Translation-based scores are not available in the mwetoolkit, but this is intended as future work.

5 Conclusions and open issues

In this chapter, we have presented an overview of research-oriented computational tools for the automatic discovery of phraseological units in corpora. We have used the mwetoolkit as an example of such tool, describing its overall architecture and exemplifying the steps of the discovery process and their results on a toy corpus of English. Given the vast literature on discovering multiword units in corpora, we believe that it is important to focus on concrete tools and examples so that new users do not feel overwhelmed by the amount of scientific articles on the topic. We believe that this chapter makes a step in this direction.

The mwetoolkit is far from being perfect, and can be improved in many ways. The first and obvious limitation is the use of terminal commands rather than a visual interface. Providing its functionalities as a web application would break the access barrier for users who are not familiar with typing commands in a Unix prompt. The development of a computational tool for phraseological discovery that is both free of charge (e.g. mwetoolkit) and easy to use (e.g. Sketch Engine) should be one of the priorities for computational linguistics.

Building better tools to support the construction of phraseological resources has a potential benefit to computational tools themselves. For example, a lexicon containing multiword units can be integrated into tools that perform automatic syntactic and/or semantic analysis of texts (Savary et al., 2017). Therefore, synergies between lexicography, phraseology and computational linguistics can help creating a virtuous circle in which computer engineers build better tools for lexicographers who, in turn, build better machine-readable dictionaries, with better coverage of phraseological units. The latter can then be integrated into computational linguistic software to improve their their linguistic precision.

References

(2010). *Proc. of the Seventh LREC (LREC 2010)*, Valetta, Malta. ELRA.

(2012). *Proc. of the Eigth LREC (LREC 2012)*, Istanbul, Turkey. ELRA.

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing*, pages 267–292. CRC Press, Taylor and Francis Group, Boca Raton, FL, USA, 2 edition.

Banerjee, S. and Pedersen, T. (2003). The design, implementation, and use of the Ngram Statistic Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 370–381, Mexico City, Mexico.

Baroni, M. and Bernardini, S., editors (2006). *Wacky! Working papers on the Web as Corpus.* GEDIT, Bologna, Italy. 224 p.

Bond, F., Kim, S. N., Nakov, P., and Szpakowicz, S., editors (2013). *Nat. Lang. Eng. Special Issue on Noun Compounds*, volume 19. Cambridge UP, Cambridge, UK.

Bonin, F., Dell'Orletta, F., Montemagni, S., and Venturi, G. (2010). A contrastive approach to multi-word extraction from domain-specific corpora. In (con, 2010).

Carpuat, M. and Diab, M. (2010). Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In *Proc. of HLT: The 2010 Annual Conf. of the NAACL (NAACL 2003)*, pages 242–245, Los Angeles, California. ACL.

Church, K. and Hanks, P. (1990). Word association norms mutual information, and lexicography. *Comp. Ling.*, 16(1):22–29.

Constant, M., Roux, J. L., and Sigogne, A. (2013). Combining compound recognition and PCFG-LA parsing with word lattices and conditional random fields. *ACM Trans. Speech and Lang. Process. Special Issue on MWEs: from theory to practice and use, part 2 (TSLP)*, 10(3).

Constant, M. and Tellier, I. (2012). Evaluating the impact of external lexical resources into a CRF-based multiword segmenter and part-of-speech tagger. In (con, 2012).

Cook, P. and Stevenson, S. (2010). Automatically identifying the source words of lexical blends in English. *Comp. Ling.*, 36(1):129–149.

Cordeiro, S., Ramisch, C., Idiart, M., and Villavicencio, A. (2016a). Predicting the compositionality of nominal compounds: Giving word embeddings a hard time. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1986–1997. Association for Computational Linguistics.

Cordeiro, S., Ramisch, C., and Villavicencio, A. (2016b). Mwetoolkit+sem: Integrating word embeddings in the mwetoolkit for semantic mwe processing. In *LREC 2016*, Portoroz, Slovenia.

Dagan, I. and Church, K. (1994). Termight: Identifying and translating technical terminology. In *Proc. of the 4th ANLP Conf. (ANLP 1994)*, pages 34–40, Stuttgart, Germany. ACL.

Daille, B. (1995). Repérage et extraction de terminologie par une approche mixte statistique et linguistique. *Traitement Automatique des Langues*, 1-2:101–118.

de Medeiros Caseli, H., Villavicencio, A., Machado, A., and Finatto, M. J. (2009). Statistically-driven alignment-based multiword expression identification for technical domains. In Anastasiou, D., Hashimoto, C., Nakov, P., and Kim, S. N., editors, *Proc. of the ACL Workshop on MWEs: Identification, Interpretation, Disambiguation, Applications (MWE 2009)*, pages 1–8, Suntec, Singapore. ACL.

Drouin, P. (2004). Detection of domain specific terminology using corpora comparison. In *Proc. of the Fourth LREC (LREC 2004)*, Lisbon, Portugal. ELRA.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Comp. Ling.*, 19(1):61–74.

Duran, M. S. and Ramisch, C. (2011). How do you feel? investigating lexical-syntactic patterns in sentiment expression. In *Proceedings of Corpus Linguistics 2011: Discourse and Corpus Linguistics Conference*, Birmingham, UK.

Duran, M. S., Ramisch, C., Aluísio, S. M., and Villavicencio, A. (2011). Identifying and analyzing Brazilian Portuguese complex predicates. In (Kordoni et al., 2011), pages 74–82.

Duran, M. S., Scarton, C. E., Aluísio, S. M., and Ramisch, C. (2013). Identifying pronominal verbs: Towards automatic disambiguation of the clitic 'se' in Portuguese. In (Kordoni et al., 2013), pages 93–100.

Evert, S. (2004). *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, Stuttgart, Germany. 353 p.

Evert, S. and Krenn, B. (2005). Using small random samples for the manual evaluation of statistical association measures. *Comp. Speech & Lang. Special issue on MWEs*, 19(4):450–466.

Fazly, A., Cook, P., and Stevenson, S. (2009). Unsupervised type and token identification of idiomatic expressions. *Comp. Ling.*, 35(1):61–103.

Finlayson, M. and Kulkarni, N. (2011). Detecting multi-word expressions improves word sense disambiguation. In (Kordoni et al., 2011), pages 20–24.

Ha, L. A., Fernandez, G., Mitkov, R., and Corpas, G. (2008). Mutual bilingual terminology extraction. In *Proc. of the Sixth LREC (LREC 2008)*, Marrakech, Morocco. ELRA.

Heid, U. (2008). *Phraseology. An interdisciplinary perspective*, chapter Computational phraseology. An overview, pages 337–360. Jhohn Benjamins, Amsterdam, Netherlands.

Heid, U., Fritzinger, F., Hinrichs, E., Hinrichs, M., and Zastrow, T. (2010). Term and collocation extraction by means of complex linguistic web services. In (con, 2010).

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., and Suchomel, V. (2014). The sketch engine: ten years on. *Lexicography*, 1(1):7–36.

Köper, M., Schulte im Walde, S., Kisselew, M., and Padó, S. (2016). Improving zero-shot-learning for german particle verbs by using training-space restrictions and local scaling. In *Proceedings of *SEM 2016*, pages 91–96. ACL.

Kordoni, V., Ramisch, C., and Villavicencio, A., editors (2011). *Proc. of the ACL Workshop on MWEs: from Parsing and Generation to the Real World (MWE 2011)*, Portland, OR, USA. ACL.

Kordoni, V., Ramisch, C., and Villavicencio, A., editors (2013). *Proc. of the 9th Workshop on MWEs (MWE 2013)*, Atlanta, GA, USA. ACL.

Linardaki, E., Ramisch, C., Villavicencio, A., and Fotopoulou, A. (2010). Towards the construction of language resources for Greek multiword expressions: Extraction and evaluation. In Piperidis, S., Slavcheva, M., and Vertan, C., editors, *Proc. of the LREC Workshop on Exploitation of multilingual resources and tools for Central and (South) Eastern European Languages*, pages 31–40, Valetta, Malta. May.

Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing.* MIT Press, Cambridge, USA. 620 p.

Markantonatou, S., Ramisch, C., Savary, A., and Vincze, V., editors (2017). *Proc. of the 13th Workshop on MWEs (MWE 2017)*, Valencia, Spain. ACL.

Martens, S. and Vandeghinste, V. (2010). An efficient, generic approach to extracting multi-word expressions from dependency trees. In Laporte, É., Nakov, P., Ramisch, C., and Villavicencio, A., editors, *Proc. of the COLING Workshop on MWEs: from Theory to Applications (MWE 2010)*, pages 84–87, Beijing, China. ACL.

McKeown, K. R. and Radev, D. R. (1999). Collocations. In Dale, R., Moisl, H., and Somers, H., editors, *A Handbook of Natural Language Processing*, chapter 15, pages 507–523. Marcel Dekker, New York, NY, USA.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Morin, E. and Daille, B. (2010). Compositionality and lexical alignment of multi-word terms. *Lang. Res. & Eval. Special Issue on Multiword expression: hard going or plain sailing*, 44(1-2):79–95.

Morin, E., Daille, B., Takeuchi, K., and Kageura, K. (2007). Bilingual terminology mining - using brain, not brawn comparable corpora. In *Proc. of the 45th ACL (ACL 2007)*, pages 664–671, Prague, Czech Republic. ACL.

Nakov, P. and Hearst, M. A. (2005). Search engine statistics beyond the *n*-gram: Application to noun compound bracketing. In Dagan, I. and Gildea, D., editors, *Proc. of the Ninth CoNLL (CoNLL-2005)*, pages 17–24, University of Michigan, MI, USA. ACL.

Pearce, D. (2001). Synonymy in collocation extraction. In *WordNet and Other Lexical Resources: Applications, Extensions and Customizations (NAACL 2001 Workshop)*, pages 41–46.

Pecina, P. (2008). *Lexical Association Measures: Collocation Extraction.* PhD thesis, Faculty of Mathematics and Physics, Charles University. 143 p.

Pecina, P. (2011). Syntax-based collocation extraction violeta seretan (university of geneva) berlin: Springer (text, speech and language technology series, volume 44), 2011, xi+217 pp; hardbound, ISBN 978-94-007-0133-5, $139.00. *Comp. Ling.*, 37(3):631–633.

Pedersen, T., Banerjee, S., McInnes, B., Kohli, S., Joshi, M., and Liu, Y. (2011). The *n*-gram statistics package (text::NSP) : A flexible tool for identifying *n*-grams, collocations, and word associations. In (Kordoni et al., 2011), pages 131–133.

Ramisch, C. (2015). *Multiword Expressions Acquisition: A Generic and Open Framework*, volume XIV of *Theory and Applications of Natural Language Processing.* Springer.

Ramisch, C., Araujo, V. D., and Villavicencio, A. (2012). A broad evaluation of techniques for automatic acquisition of multiword expressions. In *Proc. of the ACL 2012 SRW*, pages 1–6, Jeju, Republic of Korea. ACL.

Ramisch, C., Schreiner, P., Idiart, M., and Villavicencio, A. (2008). An evaluation of methods for the extraction of multiword expressions. In Grégoire, N., Evert, S., and Krenn, B., editors, *Proc. of the LREC Workshop Towards a Shared Task for MWEs (MWE 2008)*, pages 50–53, Marrakech, Morocco.

Ramisch, C., Villavicencio, A., and Boitet, C. (2010a). Multiword expressions in the wild? the mwetoolkit comes in handy. In Liu, Y. and Liu, T., editors, *Proc. of the 23rd COLING (COLING 2010) — Demonstrations*, pages 57–60, Beijing, China. The Coling 2010 Organizing Committee.

Ramisch, C., Villavicencio, A., and Boitet, C. (2010b). mwetoolkit: a framework for multiword expression identification. In (con, 2010), pages 662–669.

Ramisch, C., Villavicencio, A., and Boitet, C. (2010c). Web-based and combined language models: a case study on noun compound identification. In Huang, C.-R. and Jurafsky, D., editors, *Proc. of the 23rd COLING (COLING 2010) — Posters*, pages 1041–1049, Beijing, China. The Coling 2010 Organizing Committee.

Ramisch, C., Villavicencio, A., and Kordoni, V., editors (2013). *ACM Trans. Speech and Lang. Process. Special Issue on MWEs: from theory to practice and use, part 1 (TSLP)*, volume 10. ACM, New York, NY, USA.

Rayson, P., Piao, S., Sharoff, S., Evert, S., and Moirón, B. V., editors (2010). *Lang. Res. & Eval. Special Issue on Multiword expression: hard going or plain sailing*, volume 44. Springer.

Riedl, M. and Biemann, C. (2013). Scaling to large3 data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 884–890. Association for Computational Linguistics.

Riedl, M. and Biemann, C. (2015). A single word is not enough: Ranking multiword expressions using distributional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2430–2440. Association for Computational Linguistics.

Rivera, O. M., Mitkov, R., and Pastor, G. C. (2013). A flexible framework for collocation retrieval and translation from parallel and comparable corpora. In Mitkov, R., Monti, J., Pastor, G. C., and Seretan, V., editors, *Proc. of the MT Summit 2013 MUMTTT workshop (MUMTTT 2013)*, pages 18–25, Nice, France.

Roller, S., im Walde, S. S., and Scheible, S. (2013). The (un)expected effects of applying standard cleansing models to human ratings on compositionality. In (Kordoni et al., 2013), pages 32–41.

Sag, I., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *Proc. of the 3rd CICLing (CICLing-2002)*, volume 2276/2010 of *LNCS*, pages 1–15, Mexico City, Mexico. Springer.

This is a pre-print of an article published in *Computational Phraseology* by John Benjamins Publishers.
The final authenticated version is available online at: https://doi.org/10.1075/ivitra.24.06ram

Salehi, B. and Cook, P. (2013). Predicting the compositionality of multiword expressions using translations in multiple languages. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 266–275. Association for Computational Linguistics.

Salehi, B., Cook, P., and Baldwin, T. (2015). A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983. Association for Computational Linguistics.

Sangati, F., Zuidema, W., and Bod, R. (2010). Efficiently extract rrecurring tree fragments from large treebanks. In (con, 2010).

Savary, A., Ramisch, C., Cordeiro, S., Sangati, F., Vincze, V., QasemiZadeh, B., Candito, M., Cap, F., Giouli, V., Stoyanova, I., and Doucet, A. (2017). The parseme shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain. ACL.

Schneider, N., Onuffer, S., Kazour, N., Danchik, E., Mordowanec, M. T., Conrad, H., and Smith, N. A. (2014). Comprehensive annotation of multiword expressions in a social web corpus. In *Proc. of the Ninth LREC (LREC 2014)*, Reykjavik, Iceland. ELRA.

Seretan, V. (2011). *Syntax-Based Collocation Extraction*, volume 44 of *Text, Speech and Language Technology*. Springer, Dordrecht, Netherlands, 1st edition. 212 p.

Seretan, V. and Wehrli, E. (2009). Multilingual collocation extraction with a syntactic parser. *Lang. Res. & Eval. Special Issue on Multilingual Language Resources and Interoperability*, 43(1):71–85.

Silva, J. and Lopes, G. (1999). A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In *Proceedings of the Sixth Meeting on Mathematics of Language (MOL6)*, pages 369–381, Orlando, FL, USA.

Smadja, F. A. (1993). Retrieving collocations from text: Xtract. *Comp. Ling.*, 19(1):143–177.

Tsvetkov, Y. and Wintner, S. (2011). Identification of multi-word expressions by combining multiple linguistic information sources. In Barzilay, R. and Johnson, M., editors, *Proc. of the 2011 EMNLP (EMNLP 2011)*, pages 836–845, Edinburgh, Scotland, UK. ACL.

Vargas, N., Ramisch, C., and Caseli, H. (2017). Discovering light verb constructions and their translations from parallel corpora without word alignment. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 91–96, Valencia, Spain. ACL.

Villavicencio, A., Bond, F., Korhonen, A., and McCarthy, D., editors (2005). *Comp. Speech & Lang. Special issue on MWEs*, volume 19. Elsevier.

Villavicencio, A., Kordoni, V., Zhang, Y., Idiart, M., and Ramisch, C. (2007). Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In Eisner, J., editor, *Proc. of the 2007 Joint Conference on EMNLP and Computational NLL (EMNLP-CoNLL 2007)*, pages 1034–1043, Prague, Czech Republic. ACL.

Weller, M. and Heid, U. (2012). Analyzing and aligning German compound nouns. In (con, 2012).

Zhou, X., Zhang, X., and Hu, X. (2007). Dragon toolkit: Incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - ICTAI 2007*, volume 2, pages 197–201, Washington, DC, USA. IEEE Computer Society.