FEDERAL UNIVERSITY OF RIO GRANDE DO SUL
INFORMATICS INSTITUTE
BACHELOR OF COMPUTER SCIENCE

JOHN CRISTIAN BORGES GAMBOA

# Automatic Compositionality Detection from Corpora

Monograph presented in partial fulfillment
of the requirements for the degree of
Bachelor of Computer Science

Profa. Dra. Aline Villavicencio
Advisor

Dra. Muntsa Padró
Coadvisor

Porto Alegre, december 2013

**CIP – CATALOGING-IN-PUBLICATION**

John Cristian Borges Gamboa,

Automatic Compositionality Detection from Corpora /

John Cristian Borges Gamboa. – Porto Alegre: Graduação em Ciência da Computação da UFRGS, 2013.

52 f.: il.

Monograph – Federal University of Rio Grande do Sul. Bachelor of Computer Science, Porto Alegre, BR–RS, 2013. Advisor: Aline Villavicencio; Coadvisor: Muntsa Padró.

I. Villavicencio, Aline. II. Padró, Muntsa. III. Título.

# CONTENTS

# ABSTRACT

Phrasal verbs in English present varying levels of semantic idiosyncrasies. Aiming to detect some of these idiosyncrasies (in this case, how much of the meaning of a phrasal verb can be extracted from each of its words) a set of measures was proposed by MCC (2003), which use a thesaurus as input. This work reimplements those measures, focusing on checking how robust they are, by applying them on several thesauri. The thesauri were built using the method in Lin (1998).

We evaluate our results using a gold standard, and the results suggest the PMI as the best way to filter the contexts the verbs are found in.

**Keywords:** Multiword Expressions, Natural Language Processing, Verb-Particle Constructions, Compositionality Detection.

**Detecção Automática de Composicionalidade a partir de Corpora**

# RESUMO

A classe de verbos frasais da língua inglesa apresenta níveis variáveis de idiosincrasias semânticas. Com o objetivo de detectar algumas dessas idiossincrasias (nesse caso, quanto do significado de um verbo frasal pode ser extraído de cada uma de suas palavras) um conjunto de medidades foi proposto por MCC (2003), o qual usa um tessauro como entrada. Este trabalho reimplementa essas medidas, com o foco de verificar o quão robustas elas são, ao aplicá-las em diferentes tessauros. Os tessauros são construídos usando o método em Lin (1998).

Nós avaliamos nossos resultados usando uma *gold standard*, e os resultados sugerem o PMI como a melhor forma de filtrar os contextos nos quais os verbos são encontrados.

**Palavras-chave:** Expressões Multipalavra, Processamento de Linguagem Natural, Construções Verbo-Partícula, Detecção de Composicionalidade .

*"They gave it me," Humpty Dumpty continued, "for an un-birthday present."*
*"I beg your pardon?" Alice said with a puzzled air.*
*"I'm not offended," said Humpty Dumpty.*
*"I mean, what is an un-birthday present?"*
*"A present given when it isn't your birthday, of course."*
— LEWIS CARROL, 1871

# ACKNOWLEDGEMENTS

I thank my family, for all the support and for being along with me throughout my passage by this course. I thank the Federal University of Rio Grande do Sul for the education and my advisor for guiding me through the world of Natural Language Processing. I also thank Muntsa Padró, for the help on understanding most of the concepts I use in this thesis.

I also would like to thank the support of projects Capes-Cofecub 707/11, CNPq 482520/2012-4, 478222/2011-4, 551964/2011-1.

# LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| NLP | Natural Language Processing |
| MWE | Multiword Expression |
| VPC | Verb-Particle Construction |
| MT | Machine Translation |
| FEI | Fixed Expression (including idiom) |
| PV | Phrasal Verb |
| BNC | British National Corpus |
| PMI | Pointwise Mutual Information |
| LMI | Lexicographer’s Mutual Information |
| AM | Association Measure |
| POS | Part of Speech |

# LIST OF FIGURES

# LIST OF TABLES

# 1 INTRODUCTION

This chapter introduces this bachelor thesis, which implements compositionality detection of Verb-Particle Constructions (VPC) using corpora. In the following sections, we list some motivations to the study of Multiword Expressions (MWE), and define our objectives. Finally, we detail the organization of this work.

## Motivation

Among the natural languages in the world, it is not difficult to find terms whose meaning depends on the joint presence of one or more terms. For instance, in English, *e.g.* and *ad hoc* are both two-word expressions whose sense is not extractable through that of each one of their words.

Though the above examples are fixed expressions in English, treating Multiword Expressions (MWE) as if they were one word only is often not enough. Other kinds of MWE could present lexical variations such as, for example, inflections according to number (e.g., *car parks*). These inflections do not necessarily occur in the last word of the expression, often appearing at the end of some other word (e.g., *parts of speech*).

Even more defying are some multiword verbs: how would one deal, for example, with verbs like *give up*, whose object could appear either before or after the particle *up*? (e.g., "I *give* it *up*" and "I *gave up* this bachelor thesis already") Or how would one distinguish the meaning of the verb *make* in sentences like "these products make up 30% of the market" from that in "make this exercise up to three times". While it could be obvious for a human to perceive such differences, a computational system may have problems in distinguishing when there is a relation between the verb and the particle from when there is not.

Researching the field of MWE is therefore of great benefit to many NLP applications. Machine Translation (MT) systems would benefit in that expressions like *kick the bucket* (meaning *die*) would not anymore be translated literally BOU (2012). Search engines would have their results improved by finding single word synonyms for MWE searches. Keyphrase extraction – an inherently multiword task – could also improve search engines results by providing more information with which to compare to the searched words (KIM 2013).

This work focuses on the multiword verbs referred above. We have implemented a set of measures aiming to automatically find out how much of their meaning we can extract by examining separately each one of their words, distinguishing cases like *carry the bags up*, whose meaning can be inferred from the meaning of the individual words, from *trip the light fantastic*, where it is impossible to do such an analysis.

## Objectives

Our work is based on (MCC 2003), who proposed a set of measures for detecting how compositional a Verb-Particle Construction (VPC) is. As input for their measures, they used a distributional thesaurus composed by verbs built as described by LIN (1998), under the hypothesis that words used with similar meanings tend to have similar neighborhoods.

In our work, we test how robust these measures are with different thesauri built by a variety of methods. Evaluation is done by measuring the correlation with a gold standard.

The results obtained suggest that improvement can be produced by filtering the contexts the verbs are found in by their highest PMI.

## Organization of this Work

This work is organized as follows: in Chapter 2, we focus on defining what Multiword Expressions are and list the different groups in which they can be classified. Additionally, we discuss how this work fits among the related work.

A detailed explanation of the implementation of this work (the method used to measure the compositionality of VPCs) is presented in Chapter 3. Chapter 4 discusses the results produced through this implementation, and Chapter 5 concludes, also pointing out what we could do next.

# 2 MULTIWORD EXPRESSIONS

This chapter discusses the topic of Multiword Expressions (MWE) as a whole. We start by defining MWE and then proceed to common properties found in the literature. A classification is then presented, followed by a discussion of related work.

## Defining MWEs

Multiword Expressions are a very frequent phenomenon. The English language, focus of this work, is very rich in such expressions. JAC (1997) uses a game called *Wheel of Fortune* as an example of how frequent they are. In the game, people are challenged to guess words and phrases that are often idioms, proverbs, famous people's names etc. He argues that, although the game is already on the air for over ten years (six days a week), there is no worry about running out of puzzles or having to start repeating them. Passing by several other classes of expressions composed by more than only one word that we store in our minds as a "compound" (family names, colleagues, neighbors, lyrics, poetry, ...), instead of separately, he then estimates that they are so frequent that "their number is of about the same order of magnitude as the single words of the vocabulary".

There is no universally agreed definition on the term Multiword Expression (RAY 2010). In this text, we adopt the following definition (BAL 2010):

> Multiword expressions (MWEs) are lexical items that: (a) can be decomposed into multiple lexemes; and (b) display lexical, syntactic, semantic, pragmatic and/or statistical idiomaticity

According to this definition, compounds like *Computer Science*, *traffic light*, or *colon cancer tumor suppressor protein* are considered to be MWE, as well as verbs like *make up* and *shoot the boot*. Light verb constructions like *take a walk* or *give a demo* are also considered MWE, as well as collocations like *seldom ever*, *strong tea*, or *powerful computer*. Finally, idioms like *Achilles' heel* or *sleep with the fishes* and fixed expressions like *ad hoc*, *e.g.* and *in short* are also considered as such.

Note that the criteria for (a) varies from language to language. For example, because of the high productivity of word concatenation rules in German, it is possible that a MWE is composed by only one word (in the sense that there are no spaces separating two lexical items). For example, the word *Arbeitgeber* in German is composed by the concatenation of *Arbeit* (work) and *geber* (giver, donor). In English, we will not concentrate on cases like "*nutshell*", "*snowflake*" or "*moonlight*", as these are naturally treated as single word, and concentrate instead on sequences of two or more non-adjacent words.

It is noticeable by the definition that a term must present some degree of idiomaticity to be considered a MWE. MWEs often present simultaneously multiple types of such idiomaticities, as can be seen in Table 2.1 (as in (BAL 2010)).

| | Lexical | Syntatic | Semantic | Pragmatic | Statistical |
|---|---|---|---|---|---|
| all aboard | - | - | - | + | + |
| bus driver | - | - | + | - | + |
| by and large | - | + | + | - | + |
| kick the bucket | - | - | + | - | + |
| look up | - | - | + | - | + |
| shoch and awe | - | - | - | + | + |
| social butterfly | - | - | + | - | + |
| take a walk | - | - | + | - | ? |
| to and fro | ? | - | - | - | + |
| traffic light | - | - | + | - | + |
| eat chocolate | - | - | - | - | - |

Table 2.1: Classification of MWEs in Terms of Their Idiomaticity, from (BAL 2010), Table 12.2

A deepier explanation of each of the five types of idiomaticity listed in the definition is presented in the following sections, using examples from (BAL 2010).

### Lexical Idiomaticity

Lexical idiomaticity occurs when some of the components of a MWE are not part of the language vocabulary. For example, *ad hoc* is an expression where both components are not part of the English lexicon, but instead come from Latin and are not often understood by a standard English speaker. Other examples include French expressions often used in English, such as *bon appétit*, *bon voyage* and *au revoir*.

### Syntactic Idiomaticity

Syntactic idiomaticity occurs when the composing words of the MWE deviate from their common behavior. For example, while *by and large* is composed by a preposition, an adjective and a coordination between them, it is used as an adverb. Expressions like *all of a sudden* and *at first* constitute other examples of type of idiomaticity[1].

### Semantic Idiomaticity

Semantic idiomaticity occurs when the meaning of the MWE is not conveyed by the composition of the meanings of each of its terms. For example, the expression *kick the*

[1] Examples taken from http://ww2.cs.mu.oz.au/~tim/pubs/altss2004.pdf.

*bucket* (meaning *die*) is not derivable by the words *kick*, *the* and *bucket*; on the other hand, the meaning of *take a walk* is approximately extractable by each of its parts, indicating that it is not a semantically idiomatic MWE.

There are some MWEs for which it is not that easy to decide about their semantic idiomaticity. The expression *bus driver* is an example where, although both *bus* and *driver* are used in their expected meanings, there is a default expectation that the *bus driver* is "one who drives a bus", and not "one who drives *like* a bus" or "an object for driving buses with".

The concept of semantic idiomaticity is tightly related to that of *semantic compositionality*, defined as "the degree to which the phrasal meaning, once known, can be analyzed in terms of the contributions of the idiom parts." NUN (1994) In other words, it refers to how easy it is to extract the meaning of an expression based on that of each one of its words. As an example, consider the verb *drop by*. Although it has some degree of semantic idiomaticity, it is easy to infer its meaning by analysing separately each of the words *drop* and *by*. This kind of analysis is not possible for the verb *come up*, for instance. Therefore, we consider *drop by* a compositional verb, while *come up* is not compositional at all.

### Pragmatic Idiomaticity

Pragmatic idiomaticity is related to how the use of a MWE is bound to a certain situation or context. For example, while the expression *good morning* could be used ironically to address a person who slept too much, it is normally associated with mornings. Other examples include greetings like *nice to meet you*, orders like *all aboard*, and even warnings like *be right back*.

### Statistical Idiomaticity

Statistical idiomaticity occurs when the components of a MWE have some kind of "affinity", in the sense that they occur more often than synonym alternatives would. As an example[2], consider the expression *strong tea*. While both *powerful* and *strong* could be considered synonyms, a combination of the first with the word *tea* would not be allowed. On the other hand, *powerful* would be the preferred word to combine with *computer*. Both of these alternatives are possible according to the syntatic and semantic rules of the language but only one of them is the preferred form adopted by the community.

## Types of MWEs

SAG (2001) suggest a very popular typology that has been adopted in many related works. Broadly, they classify MWEs into two big groups: **Lexicalized Phrases** and **Institutionalized Phrases**. The following subsections describe each of these groups and their subgroups.

[2] Example taken from the Wikipedia: http://en.wikipedia.org/wiki/Collocation.

### Lexicalized Phrases

Lexicalized Phrases are divided in three categories: **fixed expressions**, **semi-fixed expressions** and **syntactically-flexible expressions**. They present varying levels of rigidity, as well as some syntactic or semantic idiosyncrasy.

#### *Fixed Expressions*

Fixed Expressions compose the MWE class whose elements could be easily treated as a single word or "words with spaces". Since they do not allow for any surface modifications, having a list with all of them would be enough for a NLP application.

Examples of such elements are *in a nutshell*, *in short*, *by the way*, and most "siamese twins" like *by and large*, *safe and sound* and *forever and ever*.

#### *Semi-Fixed Expressions*

Semi-fixed Expressions are further divided into three subcategories: **Non-Decomposable Idioms**, **Compound Nominals** and **Proper Names**. While they accept some forms of inflection, they normally present some syntactic idiosyncrasies.

- **Non-Decomposable Idioms** The word *Idiom* is often related to the semantic opaqueness of a centain expression, that is, to how difficult it is to extract its meaning by focusing on each of its words separately.

  *Semantic Decomposability*, in turn, is the ability of some idioms to be analised through their parts. For example, the idiom *let the cat out of the bag* could have a metaphorical interpretation, where "the cat" is a secret and "go out of the bag" is "to be revealed". This kind of analysis is not possible for *kick the bucket*, for example, characterizing it as *non-decomposable*. Other examples of non-decomposable idioms are *trip the light fantastic* and *shoot the breeze*.

- **Compound Nominals** Expressions like "car park" are very similar to Fixed Expressions, only they inflect for number.

- **Proper Names** Good examples of proper names are *San Francisco* or *Porto Alegre*. Depending on the context, their properties may vary greatly. For example, while U.S. sports team names are often made up of a place or organization and an appelation (for example, *the San Francisco 49ers* or *the Oakland Raiders*), sometimes it is possible to omit the organization name (resulting, in our examples, in *the 49ers* and *the Raiders*).

  Other interesting effect happens when the team name occurs as a modifier in a compound nominal: the definite determiner, part of the team name before, now refers to the entire compound (for example, *the/an [Oakland] Raiders player*).

#### *Syntactically-Flexible Expressions*

Syntactically-Flexible expressions accept a high number of variations. The following subcategories are further discussed: **Verb-Particle Constructions**, **Decomposable Idioms** and **Light Verb Constructions**.

- **Verb-Particle Constructions** Verb-Particle Constructions (VPC), also named Phrasal Verbs (PV), are composed by a verb followed by one or more particles. They can be semantically compositional, i.e., their meaning can be extracted based on their parts (examples include *come over* or *eat up*), or semantically idiosyncratic (*break up*, *brush up on*). Adverbs can often be put before the particle, separating it from the verb (*fight bravely on*).

  Some VPCs are transitive, in which case the complement can appear either before or after the preposition (*call Kim up* vs. *call up Kim*). There are VPCs, however, that accept only one of the forms (*fall off a truck* vs. ?*fall a truck off*).

  A comprehensive introduction to Verb-Particle Constructions can be found in (VIL 2003).

- **Decomposable Idioms** As opposed to Non-decomposable Idioms, Decomposable Idioms allow for a semantic analisys of their components. Examples of such idioms are *spill the beans*, *touch a nerve* and *pull strings*.

  By allowing such an analisys, they can be subject to a varying and highly unpredictable degree of syntatic variation. For example, they can undergo passivization (e.g., *nerves were touched*) or be internally modified (e.g., *pull a few strings*).

- **Light Verb Constructions** Including examples like *give a demo*, *have a conversation* or even the already mentioned *take a walk*, light verb constructions consist of a verb and a noun complement. They are highly idionsyncratic in that it is often difficult to tell what verbs combine with what nouns. While the noun is often used in the normal sense, the verb is often "empty" in its meaning, thus the name of the construction.

  Light verb constructions can be passivized (e.g., *a demo was given*), internally modified (e.g., *she took a long walk*) and even extracted (*How many demos did Kim give?*).

### Institutionalized Phrases

Institutionalized Phrases are not semantically nor syntactically idiosyncratic, but rather statiscally idiosyncratic (as explained in section Statistical Idiomaticity), in that the occurrence of such elements is greatly higher than that of alternative forms. For example, *traffic light* is always the preferred form for "a visual signal that controls the flow of traffic", and alternatives such as *vehicle lamp* or *traffic lamp* are not accepted.

## Related Work

This section presents part of the work that has already been done related to MWEs. We start by introducing the area as a whole and how it fits among the related areas. We then focus on the related work on NLP.

### Multiword Expressions

In the field of Phraseology, many words were used by different scholars to refer to similar (but not exactly the same) phenomena. GRI (2008) presents a survey on the use of the word "*phraseologism*" and points towards a better definition of the term.

*Multiword Item* appears in (MOO 98) as a superclass of *idioms*. *Idioms* are then used "to refer loosely to semi-transparent and opaque metaphorical expressions such as *spill the beans* and *burn one's candle at both ends*". The same book also defines and discusses the acronym *FEI* – Fixed Expressions (including idioms).

*Fixed Expressions* appears also in (JAC 97) as a set composed by *collocations*, *compounds*, *idioms*, *names*, *clichés*, *titles* and *quotations*. While he gives examples of all of these classes, he makes no effort in distinguishing *collocations* from *compounds* (and they seem to overlap).

As opposed to "free combinations", JES (1924) defines the word *Formula*:

> A formula may be a whole sentence or a group of words, or it may be one word, or it may be only part of a word, – that is not important, but it must always be something which to the actual speech-instinct is a unit which cannot be further analyzed or decomposed in the way a free combination can.

"*Formulaicity*" thus has been used to refer to the property of these word groups to be stored as a whole in the memory, instead of being created freely through the rules of the language.

WRA (2000) briefly discuss about the proliferation of terms and their use in different areas, which underestimate "some basic problems with the looseness of the terminology, which makes it extremely difficult to be sure when like is being compared with like". In avoiding the term *Formulaic Language*, too widespread in the literature, though not very clearly defined, WRA (2002) adopts *Formulaic Sequence*, which he defines as follows:

> a sequence, continuous or discontinuous, of words or other elements, which is, or appears to be, prefabricated: that is, stored and retrieved whole from memory at the time of use, rather than being subject to generation or analisys by the language grammar.

Finally, RAY (2010) discusses the development of a community of researchers around the term *Multiword Expression* (sometimes also called *Multiword Units*).

### Related Work In NLP

As lexical resources dedicated to MWEs are rare and static resources, there is a need for automatically identifying MWEs from corpora. A lot of work has been done on multiword identification, detection and extraction from corpora.

An evaluation of some extraction methods is presented in (RAM 2008), and in what follows we discuss some of the more relevant work.

RAM (2010a), RAM (2010b) and RAM (2012) discuss the *mwetoolkit*, a generic framework for extracting several types of MWEs. The toolkit implements a series of association measures (AMs) that have been often used to identify MWEs along with the ability to define patterns and filters to refine the results.

ZHA (2006) identify MWEs by using the World Wide Web as a corpus. The result is then used to improve a broad coverage precision grammar for English.

In terms of VPCs in particular, (BLA 2001), log-linear models are used to extract multiword verbs composed by a verb and any number of particles, which can be either prepositions or adverbs. BAL (2002) propose three methods to extract VPCs from corpora: (1) the use of a POS-taggers, (2) the use of a statistical chunker (whose purpose is, in their words, "partitioning up a text into syntatically-cohesive [. . . ] segments ('chunks')") and (3) the use of a statistical chunker, but taking into account some grammar rules. To improve the results, they then propose a hybrid approach, using information extracted by running each of the basic methods as features to train a memory-based learner. They produce high precision (0.859) and recall (0.871). A continuation of that work can be found in (BAL 2005), where mote detailed syntactic information obtained with the RASP parser is compared to the other approaches, resulting in a parser that is robust to low-frequency verbs and whose tests over the BNC produced a high F-score for both intransitive (0.749) and transitive (0.897) VPCs.

This work uses distributional thesauri (which are constructed using the method proposed by LIN (1998)) composed exclusively by verbs (including VPCs) to store the semantic relationship between pairs of verbs as a means to find the compositionality of VPCs. To build these thesauri, syntactic dependency relations are extracted from a parsed corpus. Then, for each pair of verbs, the nouns that occurred as subject and object of both of them are counted and their frequency is compared. The intuitive idea behind this method is that similar words tend to appear in similar contexts, in terms of syntactic relations like subject and object. It is interesting to note that this turns out to be another way of extracting multiword verbs. This work will be explained in more details in section Thesauri Construction.

As for detecting MWE compositionality, an earlier effort was the work of LIN (1999), who proposed a method whose intuitive idea was "that the metaphorical usage of a non-compositional expression causes it to have a different distributional characteristic than expressions that are similar to its literal meaning". He uses the differences of the values of Mutual Information of two collocations to decide if they form a compositional or idiomatic collocation.

Aiming to "put the treatment of non-compositionality in corpus-based NLP on a firm empirical footing", BAN (2003) built a gold standard for evaluating VPC compositionality models. To show its usefulness, they implemented a machine learning classifier targetting to decide over the contribution of each of the VPC parts for its meaning. They conclude in favor of the viability of using empirical methods to analyse VPC semantics.

BAL (2003) classify MWE in three distinct groups (*non-decomposable*, *idiosyncratically decomposable*, and *simple decomposable*), and use Latent Semantic Analysis (LSA) to distinguish *simple decomposable* MWEs from the other two groups. As a result, they show that, when used over English noun-noun compounds and VPCs, their method correlates moderately with WordNet occurrences of hyponymy.

A more recent method for detecting VPC compositionality is presented in (KIM 2007). Using the gold standard built by (MCC 2003) (described in detail in Chapter 3) and the dataset provided by (BAN 2003), they have constructed a classifier that uses the semantic similarity between the VPC and its simplex form to detect how compositional a VPC is.

Apart from a gold standard, MCC (2003) defined several measures which take a thesaurus as input and whose output was then compared to the gold standard. These measures are the base of this work, and will be discussed in more details in section Compositionality Measures.

# 3 DETECTING VPC COMPOSITIONALITY

We detect VPC compositionality following the method proposed by (MCC 2003). The method is divided in two independent phases.

The first phase consists in the construction of a distributional thesaurus. For our purposes, a thesaurus is a square matrix whose elements indicate the semantic similarity between two words. We enforce the similarity values to be between 0 (totally different) and 1 (totally similar), and the similarity between a verb and itself is always 1. For example, Table 3.1 shows the similarity between some verbs related to the verb *eat*. These values are not real ones: they were chosen to demonstrate that verbs more similar in meaning should have higher values of similarity.

| | eat | cook | swallow | digest | ... | learn | walk |
|---|---|---|---|---|---|---|---|
| eat | 1 | 0.6 | 0.7 | 0.85 | ... | 0.3 | 0.1 |
| cook | 0.6 | 1 | 0.65 | 0.75 | ... | 0.2 | 0.2 |
| swallow | 0.7 | 0.65 | 1 | 0.75 | ... | 0.3 | 0.15 |
| digest | 0.85 | 0.75 | 0.75 | 1 | ... | 0.7 | **0.8** |
| ... | ... | ... | ... | ... | ... | ... | ... |
| learn | 0.3 | 0.2 | 0.3 | 0.7 | ... | 1 | 0.2 |
| walk | 0.1 | 0.2 | 0.15 | **0.8** | ... | 0.2 | 1 |

Table 3.1: An example of a thesaurus matrix showing the similarities between some verbs related to *eat*.

Because the thesauri we use are constructed automatically, some of the values in the matrix may not correspond to the real semantic similarity between two verbs. As an example, Table 3.1 shows a high similarity between *walk* and *digest*. This would indicate that both *walk* and *digest* tend to occur with similar subject nouns and object nouns.

To build the thesauri, we use the RASP dependency parser (BRI 2006) that outputs a list of tuples in the form $(v, r, n)$ for each of the syntactic relations in a sentence, where $v$ is a verb related to the noun $n$ by the relation $r$. Using the tuples we calculate the similarity between each pair of verbs.

In the second phase of the method, we calculate a set of compositionality measures using the thesauri previously built. As a result, for each of the phrasal verbs we are interested in, we get a set of scores describing how compositional it is compared to the other verb alone (the simplex verb). To evaluate these measures, we compare them with those of a gold standard.

The following sections present in detail each of these steps. We start by discussing the resources we use. We then proceed to a description of the thesaurus construction phase, along with an explanation of the measures, followed by the evaluation methods and the construction of the gold standard.

## Resources Used

To build the abovementioned thesauri, we use the written portion of the 100 million word British National Corpus (BNC) (BUR 2000). The corpus includes samples from a variety of genres, including newspapers, academic research and books.

The corpus is parsed using the RASP parser (BRI 2006). The parser outputs a set of *grammatical relations* describing word dependencies for each of the sentences in the BNC. We use these relations to create triples of the form $(v, r, n)$ as described above. The next section details more deeply how these triples are used.

## Thesauri Construction[1]

This section describes the thesaurus construction method proposed by (LIN 98). We parse the BNC (BUR 2000) with the RASP parser (BRI 2006), using subject and object relations but removing those involving pronouns. A detailed description of the construction follows.

The thesauri are built under the hypothesis that words occurring in similar contexts often have similar meanings (the so-called *Distributional Hypothesis*). With this idea in mind, we use the *grammatical relations* given by the RASP parser to create triples of the form $(v, r, n)$, which represent the combination of a verb $v$ and a noun $n$ with a relation $r$ in a sentence, i.e., the context for a given verb in a given sentence. We represent the number of occurrences of a given context $(v, r, n)$ by $||v, r, n||$.

As an example, consider the following sentence:

*The woman bought a computer*

Because the RASP parser includes a lemmatizer, *bought* is treated as the past form of *buy*. Given the grammatical relations, the following tuples will be created:

*(buy, woman, subject)* , *(buy, computer, object)*

The triples are the basic units from which all the statistical relations between words can be derived. We can estimate the probability of a noun $n$ appearing for a given pair verb-relation as

$$p(n|v, r) \simeq \frac{||v, r, n||}{||v, r, *||} \tag{3.1}$$

[1] I thank Marco Idiart and Muntsa Padró for providing the thesauri I use as input to my application, as well as for all the discussion about the topic we had throughout its implementation.

where $*$ indicates a sum over all possible values of that variable, or

$$||v, r, *|| = \sum_n ||v, r, n||$$

Following the distributional hypothesis we could posit that the similarity between two different verbs is a measure of the closeness of their noun distributions.

But a possible problem of this method is that these distributions tend to be dominated by very frequent words that in general are polysemic and may combine with many verbs. LIN (1998) proposed that what has to be compared is not the relative frequency of the words but the information content of the triple measured by the Pointwise Mutual Information (PMI), which is defined by

$$\begin{aligned} PMI(v, r, n) &= \log \frac{p(v, n|r)}{p(v|r)p(n|r)} \\ &\simeq \log \frac{||v, r, n|| \cdot ||*, r, *||}{||v, r, *|| \cdot ||*, r, n||} \end{aligned} \qquad (3.2)$$

PMI indicates how the the frequency of $v$ and $n$ observed together departs from random chance, for a given relation $r$, and it eliminates spurious high correlation due to frequent words. Therefore Lin's version of the distributional hypothesis states that two words (verbs in our case) are similar if they have similar information content for all pairs $(r, n)$.

### Accounting for Noise

The output of the RASP parser could contain error. We expect to find some non-verbs that were considered verbs by the parser and constitute noise in our experiment. Because some verbs hardly appear in the corpus, we also do not have much information about the contexts in which they are normally used in the language. To account for these sources of possible errors, we remove verbs appearing less than 50 times in the corpus.

Before calculating similarity, various filters can be applied to remove possible sources of noise among the contexts of a given verb. The simplest is a low frequency filter, which removes contexts (dependency triples) that occur less than a certain threshold $th$, assuming that they are not frequent enough to be distinguished from random noise or relevant for determining similarity. We compare the use of several thresholds varying from $th$=1 to 50 counts per context.

A second set of filters is based on the relevance or salience of contexts according to a given measure. For instance, Lin only uses contexts with positive PMIs for determining similarity. BIE (2013) only use those with the highest Lexicographer's Mutual Information (LMI) values for each word, where LMI is defined as:

$$LMI = PMI \times frequency$$

To examine how effective they are we define several filters to keep just the top $p$ most relevant contexts for each word, where relevance is defined according to position in a rank computed with the following measures:

- Entropy: The idea behind sorting triples by entropy is that if a concrete combination of a relation and a noun appears among the contexts of many verbs, then it is probably not very informative. In such a case, this pair $(r, n)$ will have a high entropy. We compute the entropy of $(r, n)$ as $H(r, n) = -\sum P(v|r, n) \log P(v|r, n)$, where $P(v|r, n)$ is the probability of seing a concrete verb given this relation and noun. We sum over all verbs to compute the entropy of the relation and noun combination, which is what we will sort, in increasing order, to select the relevant contexts of a given verb.
- Frequency: sorts contexts by decreasing frequency.
- PMI (equation 3.2): for relevance as association strength, the higher the PMI the higher the relevance.
- LMI: Lexicographer's Mutual Information $LMI = PMI \times frequency$ using frequency to adjust PMI bias towards low frequent contexts, the higher the LMI the higher the relevance.

To examine the effect of these filters in the thesauri, we explore $p$ varying between 10 and 1000 most salient contexts.

## Compositionality Measures

MCC (2003) proposed a set of measures to detect how compositional a verb is when compared to other verbs in a thesaurus. To apply those measures, we rank, for each verb, all other verbs by similarity values. In our example in the beginning of the chapter, the situation of each verb is given by Table 3.2, where the synonyms of each verb are presented in a column below the target verb.

| eat | cook | swallow | digest | ... | learn | walk |
|---|---|---|---|---|---|---|
| eat(1) | cook(1) | swallow(1) | digest(1) | ... | learn(1) | walk(1) |
| digest(0.85) | digest(0.75) | digest(0.75) | eat(0.85) | ... | digest(0.7) | digest(0.8) |
| swallow(0.7) | swallow(0.65) | eat(0.7) | walk(0.8) | ... | eat(0.3) | cook(0.2) |
| cook(0.6) | eat(0.6) | cook(0.65) | cook(0.75) | ... | swallow(0.3) | learn(0.2) |
| ... | ... | ... | swallow(0.75) | ... | ... | ... |
| learn(0.3) | learn(0.2) | learn(0.3) | ... | ... | cook(0.2) | swallow(0.15) |
| walk(0.1) | walk(0.2) | walk(0.15) | learn(0.7) | ... | walk(0.2) | eat(0.1) |

Table 3.2: Verbs ordered by similarity values

For a given verb, we call the first 500 most similar verbs the *neighbors* of that verb. The compositionality measures we use consider only a verb's neighbors.

Note that, according to Table 3.2, a verb is always its neighbor (since its similarity with itself is always the maximum possible one). Because we are not interested on that similarity for detecting VPC compositionality, we manually change that to 0, thus making sure it will never appear as its own neighbor.

We now proceed to the description of each of the compositionality measures proposed by (MCC 2003). We use the word *complex* to refer to a VPC (e.g., *drop by*) and *simplex* to refer to its head verb (e.g., the simplex form of *drop by* would be *drop*).

- **overlap** We consider the top $X$ neighbors of a given VPC and its simplex form and count the size of the intersection of both sets. We used $X = 30$, 50, 100 and 500. The intuitive idea behind this measure is that highly compositional VPCs have a meaning more similar to that of its simplex form. Figure 3.1 shows an example of this measure (as in (MCC 2003), figure 1, page 5).

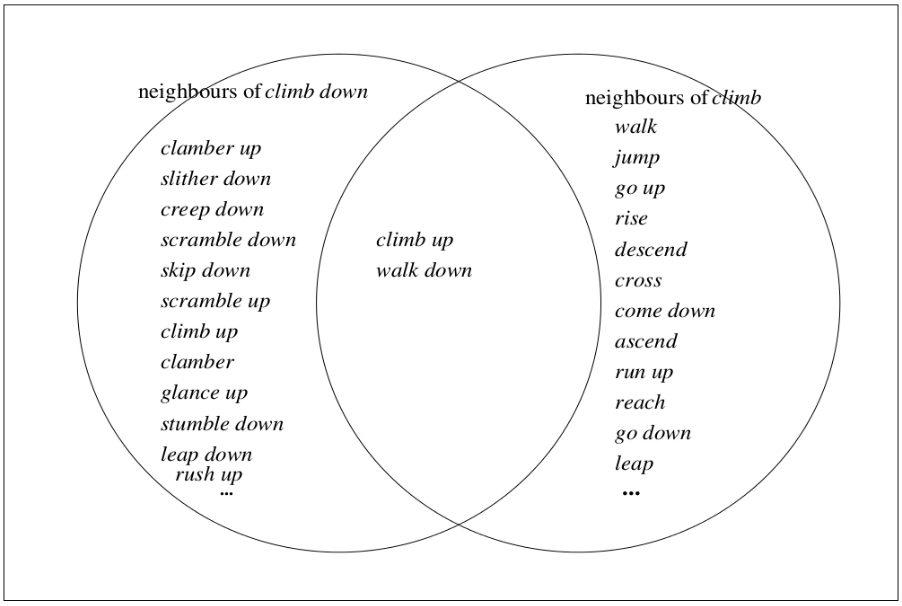

Figure 3.1: Example of the **overlap** measure, as in (MCC 2003), figure 1.

- **sameParticle** For a given VPC, we count the number of neighbors that (1) are VPCs; and (2) use its same particle. The intuition is that compositional VPCs have part of their meaning conveyed by the particle. Figure 3.2 shows an example of this measure.

- **sameParticle-simplex** For a given VPC, we count the number of neighbors that (1) are VPCs; and (2) use its same particle (i.e., the same as the previous measure). We then subtract the number of neighbors of the simplex form that use the same particle. Figure 3.3 shows an example of this measure.

- **simplexAsNeighbor** Whether the simplex verb occurs in the top 50 nearest neighbours of the complex.

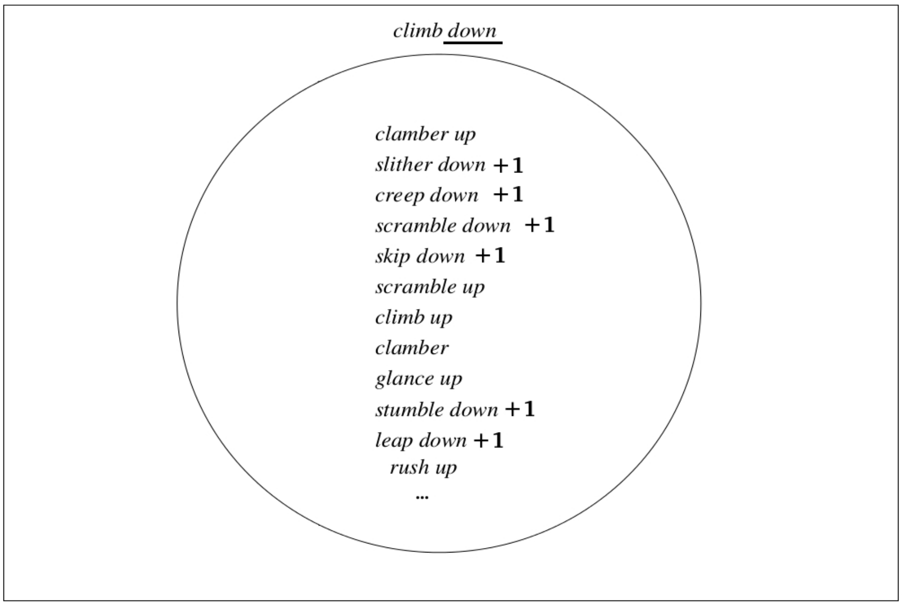

Figure 3.2: Example of the **sameParticle** measure, adapted from (MCC 2003), figure 1

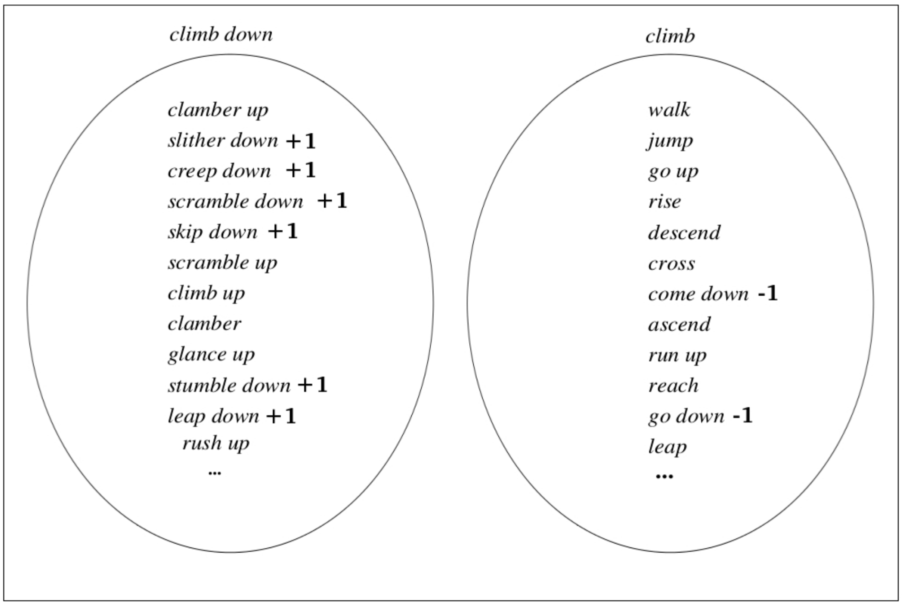

Figure 3.3: Example of the **sameParticleSimplex** measure, adapted from (MCC 2003), figure 1

- **rankOfSimplex** The rank of the simplex in the top 500 nearest neighbours of the complex.
- **scoreOfSimplex** The similarity score of the simplex in the top 500 nearest neighbours of the complex.
- **overlapS** We consider the top $X$ neighbors of a given VPC and its simplex form. For each neighbor of the complex that is a VPC, we then convert the neighbor into its simplex form. Finally, we count the size of the intersection of both sets. We used $X =$ 30, 50 and 500. As well as in the case of **sameParticle**, the intuition here is that the particle contributes part of its meaning when the verb is compositional. Figure 3.4 shows an example of this measure (as in (MCC 2003), figure 2, page 5).

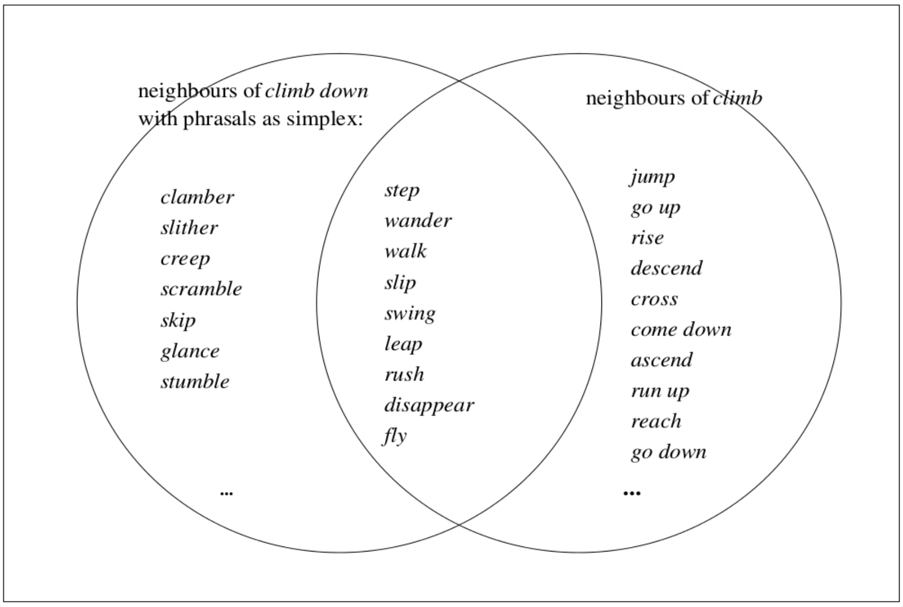

Figure 3.4: Example of the **overlapS** measure, as in (MCC 2003), figure 2.

## Gold Standard

The evaluation of the measures described in the previous section is done for a sample of VPCs of the thesaurus. MCC (2003) selected 116 VPCs and built a gold standard[2] whereby they then evaluated the measures.

3 native english speakers annotated 116 VPCs according to their semantic compositionality. The 116 verbs were divided into four groups: 34 low frequency VPCs, 33 medium frequency VPCs, 33 high frequency VPCs, and 16 manually selected VPCs.

The human annotators were then asked to assign a numerical score to each VPC according to how compositional they were: 0 indicating completely non-compositional and 10 indicating totally compositional. They could also assign a "don't know" value, in which case the verb was discarded.

[2] Available in http://mwe.stanford.edu/resources/.

From the 116 annotated verbs, only 5 were discarded. The resulting in 111 VPCs were then ranked according to the average of their three assigned scores. Table 3.3 shows an extract with the first 10 verbs of the gold standard, as annotated by each of the judges.

| VPC | Head | Particle | Frequency | Judge1 | Judge2 | Judge3 |
|---|---|---|---|---|---|---|
| call+in | call | in | 395 | 8 | 8 | 2 |
| spark+off | spark | off | 179 | 8 | 5 | 2 |
| step+out | step | out | 428 | 8 | 10 | 8 |
| come+off | come | off | 611 | 4 | 10 | 4 |
| lie+down | lie | down | 311 | 8 | 10 | 9 |
| tear+up | tear | up | 141 | 8 | 5 | 7 |
| walk+off | walk | off | 199 | 8 | 7 | 8 |
| gather+up | gather | up | 189 | 8 | 7 | 7 |
| spring+up | spring | up | 293 | 8 | 8 | 9 |
| come+up | come | up | 3145 | 8 | 10 | 5 |

Table 3.3: Extract of the Gold Standard, with the answers of the three judges.

To measure how much the three annotators agreed, MCC (2003) used the *Kendall's Coefficient of Concordance (W)*. The statistic ranges from 0 (denoting little agreement) to 1 (full agreement) and is calculated as

$$W = \frac{12\sum_{i=0}^{n} R_i^2 - 3n(n+1)^2}{n(n^2-1) - \frac{\sum_{j=1}^{k} T_j}{k}}$$

where the average rank of the iˆth item is indicated by $R_i^2$, and k is the number of judges. $T_j$ is a correction for ties, which is calculated as

$$T_j = \sum_{i=0}^{g_j} (t_i^3 - t_i)$$

where the number of tied ranks in the $i^{th}$ grouping of ranks is given by $t_i$. Calculating $k(n-1)W$, one can get a distribution which is approximated to $\chi^2$ with $n-1$ degrees of freedom. The produced $W$ was 0.594, which gives a $\chi^2$ score of 196.30, and has a probability $<= 0.000001$.

To evaluate the measures, the VPCs present in the gold standard were ranked according to the values returned by each of the measures. The Spearman rank's correlation coefficient between the ranks resulting from the annotator judgements and the ranks resulting from the compositionality measures was then calculated, telling us how well the compositionality measures predicted the real VPC compositionalities.

Because one of the measures (**simplexAsNeighbor**) is not numerical, but boolean, it was not possible to compare it to the gold standard by using the Spearman ranks correlation coefficient. Therefore, the Mann-Whitney U test was used to decide if both sets of results (i.e., the booleans resulting from applying **simplexAsNeighbor** to the thesaurus, and the average of the judgements of each VPC in the gold standard) are part of the same population.

From the application of the test, we produce a Z-Score. To compare the performance of **simplexAsNeighbor** with that of the other measures, we also calculate a Z-Score from the Spearman correlation coefficient.

# 4 RESULTS

Each of the compositionality measures is evaluated separately. The first two columns of Table 4.1 show a comparison between the results found by (MCC 2003) and the results we found when using a thesaurus built the same way they built (i.e., we wanted to reproduce the same results). The values are the Z-scores we found by comparing the output of the measures with either the Spearman's ranks correlation coefficient or with the Mann Whitney U test, as described in the previous chapter. Because we produce Z-scores, the higher their absolute values the higher the confidence that both samples are derived from the same population.

It is noticeable that the values we found are different. We used a different version of the RASP parser, causing the generated tuples to possibly differ from those generated by the version they used in (MCC 2003).

Nevertheless, the results are sufficiently similar, and the highest values for the measures in both thesauri come from the same compositionality measures.

| | **McCarthy** | **Our Results** | **Filtering low-frequency verbs** |
|---|---|---|---|
| **Number of Samples** | - | 109 | 69 |
| **overlap500** | -0.38 | -0.224 | -0.484 |
| **overlap100** | 0.39 | 0.107 | 0.636 |
| **overlap50** | 1.43 | 0.471 | 0.751 |
| **overlap30** | 1.74 | 0.816 | 0.346 |
| **sameParticle** | 4.34 | 2.884 | 2.873 |
| **sameParticle-simplex** | 5.17 | 3.564 | 3.963 |
| **simplexAsNeighbor** | 0.95 | -0.152 | -0.168 |
| **rankOfSimplex** | -1.21 | -0.129 | 0.451 |
| **scoreOfSimplex** | 0.54 | 0.032 | 0.081 |
| **overlapS30** | 3.21 | 2.146 | 2.749 |
| **overlapS50** | 3.18 | 2.583 | 2.276 |
| **overlapS500** | 1.75 | 1.679 | 0.026 |

Table 4.1: Comparison of our results and the results found by (MCC 2003).

The third column of Table 4.1 shows the values we found by removing the verbs whose frequency is less than 50, as discussed on the Thesaurus Construction section. Because we were removing the verbs for which we did not have much information, we expected the results in this column to be better than the previous ones. Despite the noticeable improvement of the **sameParticle-simplex** measure, which is not significant, there is not much improvement of the other measures. It is worth noting that from the 111 verbs from the gold standard, only 69 were found in this thesaurus, as explicited by the "Number of Samples" line in Table 4.1.

As described in the Thesaurus Construction section, we applied several filters aiming to remove possible sources of noise from the verbs in the thesauri. Table 4.2 shows the results when taking into account only the $p$ most frequent dependency tuples generated by using the RASP parser for each verb.

| **Contexts filtered by Frequency** | | | | | | |
|---|---|---|---|---|---|---|
| $p$ | **1000** | **500** | **200** | **100** | **50** | **40** |
| **Number of Samples** | 69 | 69 | 69 | 69 | 69 | 69 |
| **overlap500** | -0.736 | -0.251 | 2.206 | 3.757 | 2.561 | 2.063 |
| **overlap100** | -0.772 | 0.530 | **3.174** | **3.990** | 2.557 | 1.840 |
| **overlap50** | -0.661 | 0.978 | 2.918 | 3.564 | 2.640 | 1.735 |
| **overlap30** | -0.104 | 1.882 | 3.059 | 3.242 | 2.020 | 1.782 |
| **sameParticle** | 3.284 | 3.054 | 2.750 | 2.296 | 2.095 | 2.080 |
| **sameParticle-simplex** | **4.474** | **3.488** | 2.735 | 2.228 | **3.420** | **3.241** |
| **simplexAsNeighbor** | -0.038 | -0.019 | -0.025 | -0.025 | -0.050 | -0.062 |
| **rankOfSimplex** | 0.999 | 1.714 | 0.112 | 0.173 | 1.247 | 0.991 |
| **scoreOfSimplex** | -0.692 | -0.251 | 1.309 | 1.884 | 1.089 | 0.670 |
| **overlapS30** | 0.548 | 2.232 | 2.936 | 3.497 | 2.105 | 2.439 |
| **overlapS50** | -0.070 | 1.350 | 2.830 | 3.523 | 2.778 | 2.837 |
| **overlapS500** | -0.611 | -0.309 | 1.723 | 3.181 | 2.444 | 1.999 |

Table 4.2: Z-scores found when using thesauri built by keeping only the $p$ most frequent contexts for each verb.

By removing the less frequent contexts from each verb, we expected the quality of the information related to each verb to increase. This is not what we verify: while the best Z-score from the third column of Table 4.1 is -3.963, from the **sameParticle-simplex** measure, the best one from Table 4.2 is 4.474, from the **sameParticle-simplex** measure. Again, although there is a seemingly relevant difference between the values, this difference is not significant. The numbers in bold are the highest absolute values in each column, i.e., for each different value of $p$.

Removing too many contexts otherwise, would cause us to lack important information about the similarities between some verbs. Despite the oscillation of highest absolute values, this effect can be seen in Table 4.2, where all of the **overlap** and **overlapS** measures, from left to right, increase their Z-scores (which, we believe, means that some noise was eliminated), but later have them decreased.

Table 4.3 shows the results when using entropy as the sorting rule for selecting the $p$ most relevant contexts for each verb. Again, the highest absolute values are bold, and no improvement is made.

It is worth noting that most of the measures in Table 4.3 have their Z-score absolute value decreased when $p$ changes from 500 to 200. This would mean that there are contexts that

| **Contexts filtered by Entropy** | | | | |
|---|---|---|---|---|
| $p$ | **1000** | **500** | **200** | **100** |
| **Number of Samples** | 69 | 69 | 69 | 69 |
| **overlap500** | -1.732 | -0.871 | -0.335 | -1.081 |
| **overlap100** | 0.436 | 1.354 | 0.482 | -1.286 |
| **overlap50** | 0.983 | 1.490 | 0.717 | -1.326 |
| **overlap30** | 0.787 | 1.530 | 0.012 | -2.284 |
| **sameParticle** | 2.860 | 2.767 | 2.113 | 1.843 |
| **SameParticle-simplex** | **3.031** | **3.163** | **3.062** | **2.562** |
| **simplexAsNeighbor** | -0.032 | -0.115 | -0.174 | -0.224 |
| **rankOfSimplex** | -0.130 | -1.527 | -0.089 | 1.191 |
| **scoreOfSimplex** | -0.987 | -0.100 | -1.309 | -0.438 |
| **overlapS30** | 1.093 | 1.877 | 0.981 | -1.578 |
| **overlapS50** | 1.207 | 1.312 | 1.137 | -1.249 |
| **overlapS500** | -1.775 | -0.874 | -0.148 | -0.875 |

Table 4.3: Z-scores found when using thesauri built by keeping only the $p$ contexts with the lowest entropy.

appear with many verbs that carry important information about the semantics of the verbs.

Table 4.4 shows the results when using PMI as the sorting rule for selecting the $p$ most relevant contexts for each verb. Once again, the results in Table 4.4 outperfom those in Table 4.1. Unfortunately, a significance test shows another time that this difference is not significant.

We also notice that, except from **overlap500** and **overlap100**, the same "movement" on the results as those we had seen when sorting contexts by frequency is seen in Table 4.4, that is, from left to right, the Z-scores start by improving, but then end by worsening their values.

Finally, we show in Table 4.5 the Z-scores resulting from applying the measures in thesauri where only the $p$ contexts with the highest LMI value were kept. Because the LMI takes into account the frequency, the same effect as in Table 4.2 can once again be seen: the absolute values of the Z-scores resulting from the application of the **overlap** and **overlapS** measures increase as $p$ decreases, but at some point they turn to decrease.

Although no significant improvement on the task performance was found, there were significant changes on the results when compared to those on Table 4.1. An example of such a important change is the value returned by **overlap500**, when $p = 200$ on Table 4.5. These changes are important because they show us that the context filters really have an influence on the results.

One of the obstacles we had during the implementation of the measures was on the application of the Mann Whitney U test to compare the **simplexAsNeighbor** measure to the other ones. MCC (2003) do not explicit how they converted the boolean values of the measure so that they would be comparable to the gold standard. Our first application used 1 (true) and 0 (false) as the values returned from the measure (note that the gold standard values range from 0 to 10). The results were too different from those of MCC (2003), so we decided to convert the booleans into 10 (true) and 0 (false).

| **Context filtered by PMI** | | | | |
|---|---|---|---|---|
| $p$ | **1000** | **500** | **200** | **100** |
| **Number of Samples** | 69 | 69 | 69 | 69 |
| **overlap500** | 1.266 | 1.251 | -0.167 | -3.347 |
| **overlap100** | 1.895 | 1.492 | 0.837 | -0.542 |
| **overlap50** | 1.606 | 2.219 | 1.373 | -0.921 |
| **overlap30** | 1.111 | 1.571 | 1.387 | 0.006 |
| **sameParticle** | 2.508 | 2.374 | 1.730 | 2.834 |
| **sameParticle-simplex** | **3.193** | 2.644 | **2.577** | **5.234** |
| **simplexAsNeighbor** | -0.180 | -0.211 | -0.251 | -0.265 |
| **rankOfSimplex** | 1.762 | 0.704 | -0.796 | -0.836 |
| **scoreOfSimplex** | 0.995 | -0.030 | -1.034 | -0.772 |
| **overlapS30** | 2.795 | 3.084 | 1.722 | -0.887 |
| **overlapS50** | 2.587 | **3.310** | 1.998 | -1.656 |
| **overlapS500** | 1.220 | 1.573 | -0.134 | -3.681 |

Table 4.4: Z-scores found when using thesauri built by keeping only the $p$ contexts with the highest PMI.

| **Context filtered by LMI** | | | | | |
|---|---|---|---|---|---|
| $p$ | **1000** | **500** | **200** | **100** | **40** |
| **Number of Samples** | 69 | 69 | 69 | 69 | 69 |
| **overlap500** | 0.702 | 1.970 | **3.287** | 2.464 | -0.233 |
| **overlap100** | 1.359 | 2.403 | 2.775 | 2.225 | -0.309 |
| **overlap50** | 1.348 | 1.910 | 2.185 | 1.703 | -0.060 |
| **overlap30** | 0.756 | 0.930 | 1.653 | 1.682 | 0.322 |
| **sameParticle** | 2.830 | 2.514 | 2.481 | 2.074 | 1.566 |
| **SameParticle-simplex** | **3.917** | **3.459** | 2.787 | 2.652 | **2.642** |
| **simplexAsNeighbor** | -0.121 | -0.080 | -0.050 | -0.056 | -0.121 |
| **rankOfSimplex** | 2.291 | 2.344 | 1.801 | 0.650 | -0.141 |
| **scoreOfSimplex** | 0.709 | 0.075 | 1.306 | 1.447 | -0.633 |
| **overlapS30** | 1.950 | 2.008 | 2.859 | **2.749** | 1.190 |
| **overlapS50** | 2.212 | 2.660 | 2.931 | 2.381 | 0.942 |
| **overlapS500** | 0.925 | 2.184 | 2.997 | 2.269 | -0.428 |

Table 4.5: Z-scores found when using thesauri built by keeping only the $p$ contexts with the highest LMI.

As the results were still not satisfactory, our third option was to normalize the values of the gold standard into 10 (any value greater than 5, including 5) and 0 (any value smaller than 5). The results were the ones used in this chapter.

In sum, although using different thesauri creation methods influenced the results of the compositionality measures significantly, no significant improvement[1] on the task was found.

[1] A significance test was applied to each pair of results. By finding a Z-score from each result, we wanted to know if the difference was significant. We thus made a Z-test by calculating $test = \frac{z_1 - z_2}{standarderror}$. We concluded that a difference was significant when the $test$ was bigger than 1,96.

# 5 CONCLUSION

Aiming to check the robustness of the set of measures proposed by (MCC 2003) to detect how compositional a group of VPCs is, we have implemented and applied them onto several distributional thesauri. We then showed that applying filters on the dependency relations used during the thesauri construction phase brought no improvement on the task results.

We also discussed how the application of context filters in the thesaurus construction influences the compositionality measures we used. Despite the lack of improvement on the task performance, significant changes on the results were found.

By implementing these measures, we noticed that their returned values are highly unpredictable. Apart from the effect seen on Tables 4.2, 4.4 and 4.5 with measures **overlap** and **overlapS** discussed in the previous chapter, we could predict neither the direction nor the magnitude of the changes in any of the other cases. Therefore, we conclude that they are not very robust. Nevertheless, we believe that this task could be used as an extrinsic evaluation for the quality of distributional thesauri, and have submitted a work to EACL that uses the results from this work as such.

Finally, we think that further improvements could be found by using a larger corpus, as suggests VIL (2003) when discussing the particularly acute data sparseness problem for multiword expressions like VPCs, because of which even a 100 million word corpus could not be enough. Thus, we intend to continue this work by applying the techniques presented in this work in the ukWaC (FER 2008), a 1 billion word corpus of the English language created by extracting text from the internet.

# REFERENCES

[BAL 2010] BALDWIN, T.; KIM, S. N. Multiword expressions. In: INDURKHYA, N.; DAMERAU, F. J. (Eds.). **Handbook of natural language processing, second edition**. Boca Raton, FL: CRC Press, Taylor and Francis Group, 2010. ISBN 978-1420085921.

[BAL 2005] BALDWIN, T. Deep lexical acquisition of verb-particle constructions. **Comput. Speech Lang.**, London, UK, UK, v.19, n.4, p.398–414, Oct. 2005.

[BAN 2003] BANNARD, C.; BALDWIN, T.; LASCARIDES, A. A statistical approach to the semantics of verb-particles. In: ACL 2003 WORKSHOP ON MULTIWORD EXPRESSIONS: ANALYSIS, ACQUISITION AND TREATMENT - VOLUME 18, 2003, Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 2003. p.65–72. (MWE '03).

[BLA 2001] BLAHETA, D.; JOHNSON, M. Unsupervised learning of multi-word verbs. In: IN PROC. OF THE ACL/EACL 2001 WORKSHOP ON THE COMPUTATIONAL EXTRACTION, ANALYSIS AND EXPLOITATION OF COLLOCATIONS, 2001. **Anais...** [S.l.: s.n.], 2001. p.54–60.

[BRI 2006] BRISCOE, E.; CARROLL, J.; WATSON, R. The second release of the rasp system. In: COLING/ACL 2006 INTERACTIVE PRESENTATION SESSIONS, 2006, Sydney, Australia. **Proceedings...** [S.l.: s.n.], 2006.

[BUR 2000] BURNARD, L. **Users reference guide for the British National Corpus**. [S.l.]: Oxford University Computing Services, 2000.

[FER 2008] FERRARESI, A. et al. Introducing and evaluating ukwac, a very large web-derived corpus of english. In: IN PROCEEDINGS OF THE 4TH WEB AS CORPUS WORKSHOP (WAC-4), 2008. **Anais...** [S.l.: s.n.], 2008.

[JAC 97] JACKENDOFF, R. **The architecture of the language faculty**. [S.l.]: MIT Press, 1997. (Linguistic inquiry monographs).

[KIM 2007] KIM, S. N.; BALDWIN, T. Detecting compositionality of english verb-particle constructions using semantic similarity. In: PACLING 2007, 2007, Melbourne Australia. **Anais...** [S.l.: s.n.], 2007. p.40–48.

[KIM 2013] KIM, S. N. et al. Automatic keyphrase extraction from scientific articles. **Language Resources and Evaluation**, v.47, n.3, p.723–742, 2013.

[LIN 98] LIN, D. Automatic retrieval and clustering of similar words. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS AND 17TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS - VOLUME 2, 36., 1998, Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 1998. p.768–774. (ACL '98).

[MCC 2003] MCCARTHY, D.; KELLER, B.; CARROLL, J. Detecting a continuum of compositionality in phrasal verbs. In: IN PROCEEDINGS OF THE ACL-SIGLEX WORKSHOP ON MULTIWORD EXPRESSIONS: ANALYSIS, ACQUISITION AND TREATMENT, 2003. **Anais...** [S.l.: s.n.], 2003. p.73–80.

[MOO 98] MOON, R. **Fixed expressions and idioms in english**: a corpus-based approach. New York: Oxford University Press, 1998. (Oxford Studies in Lexicography and Lexicology).

[RAM 2008] RAMISCH, C. et al. An Evaluation of Methods for the Extraction of Multiword Expressions. In: PROCEEDINGS OF THE LREC WORKSHOP TOWARDS A SHARED TASK FOR MULTIWORD EXPRESSIONS (MWE 2008), 2008, Marrakech, Morocco. **Anais...** [S.l.: s.n.], 2008. p.50–53.

[RAY 2010] RAYSON, P. et al. Multiword expressions: hard going or plain sailing? **Language Resources and Evaluation**, v.44, n.1-2, p.1–5, 2010.

[VIL 2003] VILLAVICENCIO, A. Verb-particle constructions in the world wide web. In: IN PROCEEDINGS OF THE ACL-SIGSEM WORKSHOP ON THE LINGUISTIC DIMENSIONS OF PREPOSITIONS AND THEIR USE IN COMPUTATIONAL LINGUISTICS FORMALISMS AND APPLICATIONS, 2003. **Anais...** [S.l.: s.n.], 2003.

# APPENDIX A GOLD STANDARD

| VPC | Simplex | Particle | Frequency in BNC | Judge1 | Judge2 | Judge3 |
|---|---|---|---|---|---|---|
| call+in | call | in | 395 | 8 | 8 | 2 |
| spark+off | spark | off | 179 | 8 | 5 | 2 |
| step+out | step | out | 428 | 8 | 10 | 8 |
| come+off | come | off | 611 | 4 | 10 | 4 |
| lie+down | lie | down | 311 | 8 | 10 | 9 |
| tear+up | tear | up | 141 | 8 | 5 | 7 |
| walk+off | walk | off | 199 | 8 | 7 | 8 |
| gather+up | gather | up | 189 | 8 | 7 | 7 |
| spring+up | spring | up | 293 | 8 | 8 | 9 |
| come+up | come | up | 3145 | 8 | 10 | 5 |
| get+through | get | through | 430 | 5 | 7 | 3 |
| look+on | look | on | 187 | 8 | 3 | 7 |
| come+down | come | down | 1771 | 8 | 10 | 5 |
| fit+in | fit | in | 364 | 4 | 10 | 4 |
| seek+out | seek | out | 444 | 8 | 5 | 5 |
| come+over | come | over | 492 | 10 | 6 | 5 |
| blow+out | blow | out | 205 | 10 | 1 | 7 |
| dry+up | dry | up | 276 | 8 | 2 | 6 |
| pull+down | pull | down | 292 | 10 | 10 | 9 |
| step+down | step | down | 155 | 4 | 5 | 9 |
| look+in | look | in | 161 | 8 | 7 | 7 |
| straighten+up | straighten | up | 162 | 8 | 7 | 8 |
| stretch+out | stretch | out | 466 | 8 | 7 | 8 |
| strip+off | strip | off | 156 | 8 | 8 | 8 |
| burst+out | burst | out | 214 | 8 | 7 | 6 |
| put+out | put | out | 859 | 10 | 4 | 3 |
| point+out | point | out | 3459 | 7 | 3 | 5 |
| run+out | run | out | 1118 | 4 | 1 | 2 |
| rule+out | rule | out | 947 | 4 | 5 | 2 |
| light+up | light | up | 485 | 4 | 5 | 5 |
| look+out | look | out | 1013 | 4 | 8 | 4 |
| cut+up | cut | up | 155 | 8 | 5 | 5 |
| divide+up | divide | up | 163 | 8 | 5 | 5 |
| lead+on | lead | on | 111 | 6 | 5 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| step+off | step | off | 56 | 8 | 10 | 8 |
| move+over | move | over | 105 | 8 | 7 | 8 |
| tuck+up | tuck | up | 72 | 8 | 5 | 5 |
| queue+up | queue | up | 85 | 8 | 5 | 5 |
| rip+off | rip | off | 128 | 3 | 7 | 2 |
| cast+off | cast | off | 103 | 7 | 7 | 5 |
| ring+off | ring | off | 62 | 8 | 7 | 5 |
| cool+down | cool | down | 75 | 8 | 7 | 5 |
| play+out | play | out | 59 | 10 | 2 | 3 |
| climb+down | climb | down | 124 | 10 | 5 | 7 |
| double+up | double | up | 80 | 8 | 5 | 5 |
| clamp+down | clamp | down | 72 | 8 | 7 | 4 |
| sew+up | sew | up | 55 | 10 | 2 | 5 |
| weigh+down | weigh | down | 58 | 5 | 4 | 5 |
| lift+off | lift | off | 80 | 10 | 9 | 6 |
| fall+through | fall | through | 98 | 4 | 10 | 3 |
| tie+down | tie | down | 57 | 4 | 8 | 3 |
| tick+over | tick | over | 66 | 4 | 3 | 1 |
| hurry+up | hurry | up | 82 | 8 | 5 | 5 |
| push+on | push | on | 57 | 4 | 3 | 3 |
| thrash+out | thrash | out | 56 | 5 | 1 | 1 |
| work+in | work | in | 125 | 5 | 3 | 2 |
| lay+up | lay | up | 64 | 4 | 2 | 2 |
| arise+out | arise | out | 80 | 10 | 7 | 4 |
| wriggle+out | wriggle | out | 47 | 10 | 5 | 5 |
| stave+off | stave | off | 88 | 4 | 0 | 1 |
| trail+off | trail | off | 89 | 8 | 1 | 1 |
| drag+on | drag | on | 98 | 8 | 3 | 2 |
| sink+in | sink | in | 115 | 4 | 7 | 4 |
| trot+out | trot | out | 47 | 10 | 2 | 2 |
| clear+off | clear | off | 57 | 4 | 5 | 2 |
| see+out | see | out | 84 | 10 | 10 | 2 |
| creep+out | creep | out | 11 | 10 | 10 | 3 |
| glance+off | glance | off | 10 | 10 | 9 | 5 |
| switch+over | switch | over | 15 | 4 | 5 | 6 |
| pound+out | pound | out | 12 | 5 | 7 | 5 |
| close+off | close | off | 35 | 6 | 5 | 5 |
| latch+on | latch | on | 34 | 3 | 2 | 3 |
| cloud+over | cloud | over | 22 | 8 | 0 | 5 |
| space+out | space | out | 27 | 4 | 1 | 0 |
| talk+out | talk | out | 33 | 8 | 5 | 5 |
| lap+up | lap | up | 19 | 3 | 0 | 5 |
| whip+off | whip | off | 11 | ? | 1 | 2 |
| pack+off | pack | off | 28 | 7 | 3 | 2 |
| walk+on | walk | on | 31 | 9 | 10 | 5 |
| syphon+off | syphon | off | 14 | 8: | 1 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| blend+in | blend | in | 34 | 8 | 3 | 6 |
| clam+up | clam | up | 15 | 4 | 0 | 1 |
| grind+out | grind | out | 30 | ? | 5 | 1 |
| fly+up | fly | up | 31 | 10 | 10 | 2 |
| rush+down | rush | down | 14 | 10 | 10 | 5 |
| see+down | see | down | 12 | 10 | 10 | ? |
| advance+up | advance | up | 12 | 10 | 9 | 5 |
| spew+out | spew | out | 31 | 10 | 10 | 5 |
| book+up | book | up | 21 | 8 | 5 | 4 |
| sing+out | sing | out | 24 | 8 | 7 | 3 |
| cock+up | cock | up | 15 | 2 | 0 | 0 |
| plod+on | plod | on | 20 | 8 | 7 | 2 |
| rise+out | rise | out | 11 | 8 | 8 | 5 |
| melt+down | melt | down | 21 | 8 | 5 | 5 |
| jump+out | jump | out | 39 | 10 | 9 | 5 |
| spill+down | spill | down | 16 | 10 | 10 | ? |
| rack+up | rack | up | 18 | 5? | ? | 1 |
| feed+on | feed | on | 23 | 10 | 4 | 6 |
| head+down | head | down | 33 | 10 | 3 | 2 |
| slip+up | slip | up | 43 | 4 | 3 | 5 |
| blow+up | blow | up | 479 | 3 | 6 | 1 |
| look+up | look | up | 2345 | 4 | 3 | 1 |
| eat+up | eat | up | 100 | 8 | 5 | 5 |
| wind+up | wind | up | 394 | 4 | 2 | 5 |
| shake+off | shake | off | 217 | 4 | 3 | 7 |
| get+on | get | on | 1759 | 3 | 3 | 4 |
| spell+out | spell | out | 410 | 3 | 1 | 2 |
| bring+up | bring | up | 1262 | 3 | 3 | 2 |
| set+up | set | up | 7580 | 3 | 4 | 2 |
| grow+up | grow | up | 1555 | 5 | 6 | 5 |
| sell+out | sell | out | 266 | 3 | 3 | 1 |
| play+down | play | down | 233 | 3 | 2 | 1 |
| write+off | write | off | 314 | 2 | 2 | 1 |
| look+up | look | up | 2345 | 4 | 3 | 1 |
| pass+out | pass | out | 148 | 3 | 7 | 0 |
| pass+down | pass | down | 113 | 10 | 8 | 3 |
| carry+out | carry | out | 8498 | 3 | 2 | 1 |

# APPENDIX B RANKS BY THESAURUS

The following table lists the rank of each verb for the best compositionality measure for each thesaurus.

| | No filtering | Filtering low-frequency verbs | Filtering by Frequency | | | | | | Filtering by Entropy | | | | Filtering by PMI | | | | Filtering by LMI | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $p$ | - | - | 1000 | 500 | 200 | 100 | 50 | 40 | 1000 | 500 | 200 | 100 | 1000 | 500 | 200 | 100 | 1000 | 500 | 200 | 100 | 40 |
| measure | same particle - simplex | same Particle - simplex | same Particle - simplex | same Particle - simplex | overlap 100 | overlap 100 | same Particle - simplex | same Particle - simplex | same Particle - simplex | same Particle - simplex | same Particle - simplex | same Particle - simplex | same Particle - simplex | overlapS 50 | same Particle - simplex | same Particle - simplex | same Particle - simplex | same Particle - simplex | overlap 500 | overlap S30 | same Particle - simplex |
| advance+up | 3 | | | | | | | | | | | | | | | | | | | | |
| arise+out | 96 | 60 | 47 | 63 | 1 | 7 | 64 | 67 | 66 | 41 | 45 | 62 | 64 | 3 | 68 | 48 | 64 | 62 | 0 | 8 | 56 |
| blend+in | 76 | | | | | | | | | | | | | | | | | | | | |
| blow+out | 37 | 14 | 19 | 15 | 12 | 17 | 20 | 6 | 41 | 28 | 20 | 36 | 38 | 7 | 23 | 31 | 14 | 24 | 9 | 7 | 23 |
| blow+up | 65 | 33 | 29 | 26 | 38 | 33 | 31 | 30 | 40 | 57 | 22 | 42 | 42 | 30 | 46 | 46 | 36 | 23 | 24 | 68 | 35 |
| book+up | 36 | | | | | | | | | | | | | | | | | | | | |
| bring+up | 51 | 9 | 15 | 14 | 50 | 25 | 30 | 31 | 23 | 23 | 32 | 19 | 8 | 52 | 29 | 41 | 13 | 10 | 49 | 44 | 21 |
| burst+out | 42 | 26 | 14 | 21 | 28 | 29 | 24 | 14 | 19 | 46 | 19 | 15 | 26 | 15 | 8 | 9 | 21 | 7 | 35 | 43 | 60 |
| call+in | 61 | 32 | 18 | 19 | 22 | 23 | 11 | 13 | 34 | 27 | 37 | 35 | 30 | 68 | 24 | 33 | 23 | 22 | 27 | 55 | 50 |
| carry+out | 111 | 67 | 68 | 68 | 61 | 64 | 68 | 68 | 48 | 59 | 65 | 65 | 67 | 51 | 64 | 66 | 68 | 68 | 66 | 67 | 68 |
| cast+off | 46 | 18 | 13 | 7 | 63 | 45 | 18 | 9 | 14 | 19 | 35 | 27 | 15 | 36 | 35 | 26 | 11 | 8 | 61 | 54 | 20 |
| clam+up | | | | | | | | | | | | | | | | | | | | | |
| clamp+down | 75 | | | | | | | | | | | | | | | | | | | | |
| clear+off | 14 | | | | | | | | | | | | | | | | | | | | |
| climb+down | 5 | 24 | 10 | 11 | 9 | 9 | 59 | 62 | 8 | 11 | 6 | 6 | 27 | 19 | 55 | 6 | 24 | 46 | 16 | 32 | 59 |
| close+off | 34 | 28 | 24 | 36 | 23 | 22 | 47 | 44 | 26 | 31 | 31 | 18 | 29 | 14 | 52 | 45 | 35 | 40 | 23 | 31 | 26 |
| cloud+over | 86 | | | | | | | | | | | | | | | | | | | | |
| cock+up | 60 | | | | | | | | | | | | | | | | | | | | |
| come+down | 20 | 5 | 7 | 4 | 42 | 26 | 1 | 0 | 6 | 3 | 44 | 47 | 5 | 18 | 1 | 3 | 2 | 2 | 41 | 30 | 2 |
| come+off | 59 | 13 | 28 | 13 | 58 | 67 | 4 | 5 | 13 | 7 | 11 | 34 | 12 | 50 | 22 | 22 | 10 | 13 | 56 | 42 | 5 |
| come+over | 105 | 59 | 58 | 59 | 18 | 21 | 51 | 48 | 67 | 62 | 62 | 66 | 55 | 35 | 54 | 53 | 60 | 57 | 31 | 41 | 43 |
| come+up | 45 | 15 | 38 | 10 | 10 | 8 | 2 | 1 | 25 | 30 | 53 | 55 | 4 | 34 | 0 | 8 | 8 | 1 | 22 | 29 | 0 |
| cool+down | 69 | | | | | | | | | | | | | | | | | | | | |
| creep+out | 27 | | | | | | | | | | | | | | | | | | | | |
| cut+up | 24 | 8 | 17 | 18 | 56 | 59 | 19 | 40 | 15 | 16 | 30 | 17 | 11 | 10 | 2 | 5 | 20 | 19 | 20 | 53 | 66 |
| divide+up | 82 | 42 | 46 | 45 | 2 | 6 | 46 | 56 | 58 | 56 | 41 | 54 | 50 | 29 | 42 | 30 | 56 | 63 | 2 | 5 | 63 |
| double+up | 33 | | | | | | | | | | | | | | | | | | | | |
| drag+on | 38 | 31 | 25 | 29 | 67 | 58 | 29 | 29 | 33 | 34 | 29 | 26 | 33 | 67 | 34 | 35 | 27 | 18 | 67 | 66 | 11 |
| dry+up | 98 | 48 | 54 | 58 | 54 | 37 | 17 | 24 | 64 | 68 | 68 | 46 | 61 | 28 | 37 | 38 | 59 | 45 | 25 | 52 | 65 |
| eat+up | 31 | 17 | 23 | 25 | 32 | 28 | 38 | 12 | 29 | 4 | 3 | 3 | 14 | 13 | 7 | 2 | 12 | 35 | 36 | 12 | 42 |
| fall+through | 88 | 58 | 45 | 44 | 37 | 54 | 50 | 52 | 47 | 45 | 52 | 53 | 49 | 49 | 51 | 59 | 48 | 51 | 47 | 28 | 49 |
| feed+on | 102 | | | | | | | | | | | | | | | | | | | | |
| fit+in | 91 | 57 | 53 | 54 | 21 | 32 | 49 | 50 | 57 | 55 | 58 | 60 | 48 | 66 | 45 | 40 | 55 | 44 | 54 | 51 | 32 |
| fly+up | 7 | 4 | 4 | 1 | 25 | 19 | 5 | 8 | 3 | 2 | 0 | 0 | 2 | 27 | 5 | 1 | 1 | 4 | 13 | 27 | 3 |
| gather+up | 29 | 12 | 5 | 6 | 47 | 53 | 6 | 3 | 2 | 1 | 1 | 1 | 13 | 12 | 14 | 7 | 9 | 34 | 40 | 40 | 10 |
| get+on | 74 | 38 | 42 | 33 | 24 | 42 | 37 | 43 | 56 | 44 | 36 | 52 | 41 | 48 | 30 | 52 | 47 | 59 | 33 | 11 | 25 |
| get+through | 95 | 56 | 57 | 48 | 27 | 41 | 45 | 47 | 63 | 43 | 51 | 59 | 53 | 65 | 50 | 51 | 54 | 56 | 37 | 26 | 34 |
| glance+off | 18 | | | | | | | | | | | | | | | | | | | | |
| grind+out | 15 | | | | | | | | | | | | | | | | | | | | |
| grow+up | 68 | 23 | 27 | 30 | 31 | 44 | 23 | 39 | 24 | 18 | 25 | 41 | 25 | 64 | 10 | 37 | 33 | 39 | 50 | 39 | 41 |
| head+down | 0 | | | | | | | | | | | | | | | | | | | | |
| hurry+up | 1 | 0 | 0 | 0 | 53 | 50 | 0 | 2 | 0 | 0 | 4 | 2 | 0 | 17 | 3 | 0 | 0 | 0 | 46 | 18 | 1 |
| jump+out | 22 | 35 | 22 | 24 | 41 | 18 | 34 | 38 | 22 | 40 | 61 | 38 | 45 | 16 | 44 | 12 | 30 | 21 | 21 | 17 | 19 |
| lap+up | 112 | | | | | | | | | | | | | | | | | | | | |
| latch+on | 109 | | | | | | | | | | | | | | | | | | | | |
| lay+up | 9 | | | | | | | | | | | | | | | | | | | | |
| lead+on | 72 | 53 | 52 | 53 | 6 | 15 | 55 | 46 | 55 | 54 | 57 | 40 | 52 | 63 | 59 | 58 | 43 | 50 | 32 | 16 | 31 |
| lie+down | 21 | 10 | 9 | 12 | 36 | 38 | 10 | 20 | 21 | 26 | 28 | 21 | 7 | 33 | 13 | 15 | 6 | 6 | 52 | 38 | 12 |
| lift+off | 26 | 11 | 6 | 20 | 17 | 14 | 16 | 22 | 7 | 8 | 10 | 8 | 18 | 22 | 19 | 14 | 22 | 33 | 6 | 37 | 33 |
| light+up | 81 | 47 | 61 | 52 | 16 | 3 | 44 | 49 | 68 | 63 | 56 | 33 | 54 | 2 | 67 | 29 | 40 | 49 | 1 | 2 | 16 |
| look+in | 50 | | | | | | | | | | | | | | | | | | | | |
| look+on | 94 | 46 | 60 | 62 | 3 | 11 | 43 | 42 | 53 | 42 | 50 | 51 | 40 | 47 | 53 | 62 | 42 | 58 | 18 | 25 | 55 |
| look+out | 64 | 27 | 64 | 65 | 13 | 36 | 13 | 23 | 39 | 25 | 9 | 20 | 21 | 26 | 33 | 28 | 53 | 55 | 12 | 50 | 48 |

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| look+up | 87 | 40 | 41 | 51 | 44 | 31 | 3 | 7 | 35 | 15 | 21 | 29 | 36 | 32 | 4 | 21 | 38 | 32 | 5 | 1 | 18 |
| melt+down | 44 | | | | | | | | | | | | | | | | | | | | |
| move+over | 90 | 52 | 51 | 61 | 29 | 30 | 54 | 55 | 62 | 53 | 60 | 58 | 44 | 46 | 49 | 44 | 46 | 54 | 44 | 15 | 47 |
| pack+off | 67 | | | | | | | | | | | | | | | | | | | | |
| pass+down | 6 | 3 | 3 | 5 | 46 | 52 | 7 | 4 | 4 | 14 | 12 | 16 | 3 | 45 | 11 | 13 | 5 | 9 | 26 | 24 | 6 |
| pass+out | 10 | 2 | 1 | 2 | 40 | 40 | 9 | 11 | 1 | 13 | 24 | 14 | 1 | 62 | 9 | 20 | 4 | 3 | 53 | 65 | 4 |
| play+down | 101 | 62 | 56 | 57 | 66 | 62 | 62 | 59 | 61 | 58 | 49 | 50 | 60 | 61 | 61 | 57 | 61 | 61 | 64 | 64 | 40 |
| play+out | 58 | 22 | 50 | 60 | 34 | 35 | 61 | 36 | 38 | 33 | 27 | 10 | 20 | 44 | 18 | 19 | 58 | 65 | 28 | 23 | 38 |
| plod+on | 97 | | | | | | | | | | | | | | | | | | | | |
| point+out | 104 | 64 | 44 | 50 | 30 | 61 | 58 | 58 | 60 | 39 | 48 | 57 | 47 | 21 | 58 | 61 | 63 | 43 | 65 | 63 | 37 |
| pound+out | 28 | | | | | | | | | | | | | | | | | | | | |
| pull+down | 43 | 20 | 16 | 43 | 8 | 12 | 33 | 35 | 18 | 6 | 2 | 13 | 24 | 9 | 16 | 24 | 26 | 31 | 4 | 36 | 30 |
| push+on | 52 | | | | | | | | | | | | | | | | | | | | |
| put+out | 41 | 7 | 37 | 39 | 39 | 20 | 48 | 54 | 12 | 10 | 18 | 25 | 6 | 60 | 6 | 36 | 19 | 30 | 30 | 49 | 29 |
| queue+up | 108 | 68 | 67 | 66 | 5 | 4 | 66 | 63 | 52 | 52 | 67 | 68 | 68 | 8 | 66 | 68 | 67 | 67 | 10 | 4 | 36 |
| rack+up | 63 | | | | | | | | | | | | | | | | | | | | |
| ring+off | 57 | | | | | | | | | | | | | | | | | | | | |
| rip+off | 56 | 30 | 26 | 28 | 15 | 10 | 36 | 28 | 17 | 38 | 40 | 24 | 35 | 4 | 32 | 65 | 29 | 29 | 15 | 6 | 46 |
| rise+out | 13 | | | | | | | | | | | | | | | | | | | | |
| rule+out | 100 | 51 | 40 | 42 | 52 | 51 | 60 | 61 | 37 | 61 | 47 | 45 | 66 | 59 | 25 | 50 | 52 | 48 | 57 | 62 | 64 |
| run+out | 80 | 37 | 49 | 35 | 45 | 60 | 28 | 19 | 46 | 20 | 23 | 31 | 17 | 43 | 17 | 32 | 32 | 28 | 38 | 48 | 17 |
| rush+down | 2 | | | | | | | | | | | | | | | | | | | | |
| see+down | 4 | | | | | | | | | | | | | | | | | | | | |
| see+out | 19 | 1 | 11 | 16 | 20 | 16 | 15 | 10 | 10 | 12 | 5 | 5 | 10 | 58 | 15 | 4 | 7 | 12 | 29 | 10 | 8 |
| seek+out | 79 | 34 | 34 | 41 | 43 | 48 | 27 | 65 | 54 | 29 | 17 | 30 | 32 | 57 | 28 | 67 | 28 | 20 | 55 | 22 | 54 |
| sell+out | 78 | 43 | 66 | 67 | 19 | 27 | 65 | 53 | 43 | 32 | 26 | 23 | 39 | 31 | 43 | 56 | 66 | 66 | 19 | 14 | 28 |
| set+up | 110 | 66 | 65 | 49 | 33 | 57 | 63 | 64 | 45 | 51 | 64 | 56 | 59 | 42 | 57 | 63 | 62 | 38 | 60 | 61 | 62 |
| sew+up | 49 | | | | | | | | | | | | | | | | | | | | |
| shake+off | 73 | 36 | 33 | 38 | 64 | 66 | 22 | 27 | 32 | 22 | 16 | 12 | 28 | 41 | 48 | 39 | 18 | 25 | 63 | 47 | 45 |
| sing+out | 11 | | | | | | | | | | | | | | | | | | | | |
| sink+in | 85 | 55 | 43 | 47 | 62 | 43 | 53 | 51 | 51 | 50 | 55 | 61 | 58 | 40 | 47 | 43 | 51 | 47 | 58 | 46 | 53 |
| slip+up | 17 | | | | | | | | | | | | | | | | | | | | |
| space+out | 12 | | | | | | | | | | | | | | | | | | | | |
| spark+off | 107 | 63 | 55 | 32 | 0 | 0 | 41 | 25 | 36 | 49 | 34 | 44 | 57 | 0 | 56 | 49 | 50 | 60 | 7 | 0 | 22 |
| spell+out | 103 | 50 | 35 | 23 | 35 | 47 | 40 | 60 | 20 | 35 | 38 | 39 | 56 | 11 | 21 | 47 | 41 | 42 | 43 | 60 | 44 |
| spew+out | 113 | | | | | | | | | | | | | | | | | | | | |
| spill+down | 16 | | | | | | | | | | | | | | | | | | | | |
| spring+up | 89 | 41 | 32 | 31 | 55 | 49 | 21 | 18 | 28 | 64 | 43 | 64 | 37 | 39 | 41 | 11 | 39 | 27 | 48 | 59 | 15 |
| stave+off | 84 | | | | | | | | | | | | | | | | | | | | |
| step+down | 35 | 45 | 36 | 37 | 51 | 56 | 52 | 34 | 44 | 48 | 42 | 28 | 43 | 56 | 65 | 27 | 37 | 41 | 51 | 45 | 61 |
| step+off | 8 | 6 | 2 | 3 | 49 | 34 | 8 | 17 | 5 | 9 | 14 | 7 | 9 | 25 | 12 | 23 | 3 | 5 | 34 | 35 | 7 |
| step+out | 55 | 21 | 21 | 22 | 4 | 1 | 12 | 21 | 31 | 37 | 15 | 11 | 31 | 1 | 20 | 10 | 31 | 17 | 14 | 21 | 14 |
| straighten+up | 99 | | | | | | | | | | | | | | | | | | | | |
| stretch+out | 48 | 19 | 20 | 17 | 7 | 2 | 14 | 16 | 27 | 17 | 13 | 37 | 23 | 6 | 36 | 18 | 15 | 16 | 3 | 9 | 13 |
| strip+off | 40 | 39 | 31 | 27 | 14 | 5 | 35 | 33 | 30 | 21 | 8 | 9 | 34 | 5 | 27 | 25 | 34 | 37 | 8 | 3 | 39 |
| switch+over | 106 | | | | | | | | | | | | | | | | | | | | |
| syphon+off | | | | | | | | | | | | | | | | | | | | | |
| talk+out | 25 | | | | | | | | | | | | | | | | | | | | |
| tear+up | 66 | 44 | 59 | 64 | 60 | 55 | 57 | 41 | 59 | 66 | 46 | 43 | 63 | 24 | 40 | 64 | 57 | 26 | 45 | 34 | 58 |
| thrash+out | 39 | 29 | 30 | 34 | 68 | 68 | 32 | 45 | 16 | 47 | 54 | 32 | 16 | 55 | 31 | 34 | 25 | 15 | 68 | 58 | 57 |
| tick+over | 93 | 65 | 63 | 56 | 48 | 39 | 56 | 57 | 65 | 60 | 59 | 63 | 65 | 38 | 63 | 60 | 65 | 64 | 39 | 33 | 52 |
| tie+down | 54 | | | | | | | | | | | | | | | | | | | | |
| trail+off | 53 | 49 | 39 | 40 | 65 | 65 | 67 | 66 | 50 | 67 | 66 | 67 | 62 | 54 | 62 | 42 | 49 | 53 | 59 | 57 | 67 |
| trot+out | 70 | | | | | | | | | | | | | | | | | | | | |
| tuck+up | 32 | 25 | 12 | 9 | 26 | 24 | 25 | 15 | 9 | 24 | 33 | 22 | 19 | 23 | 39 | 16 | 17 | 11 | 17 | 13 | 9 |
| walk+off | 23 | 16 | 8 | 8 | 11 | 13 | 26 | 26 | 11 | 5 | 7 | 4 | 22 | 20 | 26 | 17 | 16 | 14 | 11 | 20 | 27 |
| walk+on | 77 | | | | | | | | | | | | | | | | | | | | |
| weigh+down | 47 | | | | | | | | | | | | | | | | | | | | |
| whip+off | 30 | | | | | | | | | | | | | | | | | | | | |
| wind+up | 83 | 61 | 62 | 55 | 57 | 46 | 39 | 37 | 49 | 65 | 63 | 49 | 51 | 37 | 60 | 55 | 45 | 36 | 42 | 19 | 24 |
| work+in | 71 | | | | | | | | | | | | | | | | | | | | |
| wriggle+out | 62 | | | | | | | | | | | | | | | | | | | | |
| write+off | 92 | 54 | 48 | 46 | 59 | 63 | 42 | 32 | 42 | 36 | 39 | 48 | 46 | 53 | 38 | 54 | 44 | 52 | 62 | 56 | 51 |