Statistical Techniques for Automatically Inferring the Semantics of Verb-Particle Constructions

Colin Bannard

School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, UK
c.j.bannard@ed.ac.uk

Abstract

This paper describes an investigation of some potential features for a statistical approach to inferring the semantics of verb-particle constructions from corpus data. Verb-particles cause particular problems for the computational semantic analysis of language, because their meaning often cannot be derived through the usual compositional methods of analysis. Two novel techniques are presented which promise to provide information about the nature and extent of composition. The first of these measures the extent to which the verb or particle of any given verb-particle may be replaced with a verb or particle of a similar semantic class to form other verb-particles that are attested in the data. The intuition here is that if it reflects systematic patterns in this way then it is more likely that the verb or particle concerned have their simplex meaning. The second technique measures the degree of semantic relatedness between the verb-particle and its component verb. The intuition here is that if a verb-particle is semantically similar to the verb then it is more likely that the verb contributes its simplex meaning. These two features are then combined and used as training data for a classifier using appropriately annotated data.

Acknowledgements

This paper has been adapted from a thesis that I submitted in partial fulfillment of requirements for the degree of MSc in Informatics at the University of Edinburgh in September 2002. The work described was carried out in the Linguistics Grammars Online laboratory at the Center for the Study of Language and Information, Stanford University, and I am indebted to all resident members of that project for their hospitality and support. My work was supervised by Tim Baldwin and am extremely grateful for his valuable advice and assistance, as well as for providing the data that was the basis of this research. I have also been grateful for the help and advice provided by Alex Lascarides. I have benefited greatly from the comments or conversation on related matters of John Beavers, Chris Callison-Burch, Ann Copestake, Dan Flickinger, Ivan Sag, Tom Wasow and Dominic Widdows. I must also thank John Carroll for providing a pre-release version of his RASP system, Marco Kuhlmann for his generous assistance when it mattered most, Diana McCarthy for making available her MDL code and Aline Villavicencio for providing the verb-particle matrix.

Table of Contents

Chapter 1

Preliminaries

This paper concerns the consideration of features that might be used for automatically inferring information about the semantics of particular kinds of multiword lexical items, namely verb-particle constructions, from corpus data. The term verb-particle construction (at least as I will be using it here) refers to combinations of a verb with an obligatory adverbial or prepositional particle (e.g. heat up, turn away, set aside). As I will explain in more detail later in this chapter, verb-particle constructions present a significant challenge to anyone attempting to implement any system for the computer processing of natural language, and particularly any system which involves semantic analysis. This paper describes the implementation and evaluation of statistical techniques that promise to be useful in supporting this task.

1.1 Defining the Verb-Particle Construction

The phenomenon that I am referring to here as the verb-particle construction has been given a variety of different names by different authors. Bolinger, 1971 (p.3) lists the following additional examples: “two-word verb”, “discontinuous verb”, “compound verb”, “verb-adverb compound”. He himself chooses the popular term “phrasal verb”. I have chosen to use “verb-particle construction” here because it seems to be the most explicitly descriptive and straightforward term available.

Deciding what exactly constitutes a verb-particle construction (hereafter referred to as VPCs) is not easy, and there is considerable disagreement within and across the above terms. I will not review the issues involved in any depth here. This is partly because it is not a topic that can be reasonably covered in a chapter like this, but principally because the decisions about what to include and what to exclude have largely

been made for me since I am working with a ready-made set of items. It will suffice to state what requirements/tests were used in identifying the data I am to use. The data that I am using for the experiments described in the paper was extracted from the Wall Street Journal Corpus, using techniques described in Baldwin and Villavicencio, 2002. This employs the following fairly coarse-grained diagnostics:

1. Transitive VPCs must allow both the joined configuration, where the verb and the particle are adjacent and the NP complement follows, or the split configuration whereby the NP complement is situated between the verb and the particle. For example, while we can say both *She looked up the article* and *She looked the article up*, thus identifying *look up* as a VPC, we can only use the construction *come with* to say *come with me* but not *come me with*. *Come with* does therefore not qualify as a VPC (it is in fact a prepositional verb).

2. Where the VPC is transitive pronominal objects must occur between the verb and the particle (in the split configuration). Consequently we can say *put them up* but not *put up them*.

3. Manner adverbs cannot occur between the verb and the particle. For example while one can *look the article up hurriedly* one cannot *look hurriedly up the article*.

I will explain how this was implemented in section 1.5.

1.2 Multiword Expressions and the Challenge for Computational Linguistics

VPCs are one particular kind of multiword expression among many (at least according to the definition of multiword expression offered by Sag et al., 2002 (p.2) as "idiosyncratic interpretations that cross boundaries (or spaces).") The issues that such items raise for various shallow processing tasks such as information retrieval or indexing have been well documented. However, I am concerned here with the problems faced by systems engaged in the linguistically precise deep processing of language, and in particular with the problems one faces in trying to produce a satisfactory semantic analysis of a sentence containing a verb particle. A number of authors have discussed the problems involved here (e.g. Pulman, 1993; Calzolari et al., 2002; Sag et al., 2002).

The discussion offered in Sag et al., 2002 is of particular interest. The authors of that paper are concerned with including mechanisms for dealing with multiword expression in a broad-coverage HPSG-based typed feature structure grammar such that it can produce an appropriate logical form for such expressions. This is the kind of task that I will have in mind throughout this paper.

Sag et al., 2002 (p.2) point out that if dealt with by "general, compositional methods of linguistic analysis", then multiword expressions will cause a few particular problems. I will discuss these here because they are very relevant to the handling of VPCs and will come up throughout this chapter. The first that they highlight is "overgeneration". Their point here is that a system that attempts to freely combine words to form multiword expressions will accept such compositional but highly improbable strings such as *powerful tea* rather than canonical phrases such as *strong tea*. While I will argue later that this is a far more pronounced with other kinds of multiword expression, this is a problem with VPCs, since most speakers would accept the sentence *Dick Turpin was strung up,* but we would not want to generate the sentence *Gulliver was strung down*. The second issue is the one that we are most concerned with here, and Sag et al., 2002 call this the "idiomaticity problem". The problem is that the meaning of a multiword expression is frequently not that we would get by combining its parts in the usual fashion. We cannot, for example, get the meaning of the sentence *They were in the outhouse making hay*, by including the simplex meanings of the verb *make* and the noun *hay*. That this is very much the case with VPCs can be seen by looking at the following sentences.

(1) We were only messing about.

(2) I'd have to look it up.

(3) Who's going to put up the money?

(4) He made it all up.

The underlying problem in each of these cases seems to be that of idiosyncrasy - that multiword expressions behave in ways that cannot be generalised from or to the rest of the language. The obvious solution then might seem to be to deal with them as completely idiosyncratic, including them item by item in the lexicon. The first problem is side-stepped since the grammar will not accept strings that are not found in the lexicon. The second problem would seem to be overcome since one can associate the idiosyncratic semantics with the particular item. There is, however, a substantial

problem with this, namely that it is impossible to provide a comprehensive list of VPCs. First of all, there are arguably as many VPCs as there are simplex verbs. And second of all, they are highly productive. That is to say that new ones can be produced, often with considerable freedom. For example one reasonably common construction is that of a denominalised verb with the particle *up*, forming constructions such as *vaccuum up*, or *box up*. It seems that one can combine certain kinds of nouns with *up* fairly freely (but not completely so), albeit to produce extremely informal sounding items.

To surmise, it appears that VPCs can not all be dealt with using the usual compositional mode of analysis, but also that we cannot approach many of them by using an inventory of forms. What, then, are we to do? I will say more about this over the next few sections, but it seems to be the case that we need to deal with some as compositional, some as not compositional, and, as I will show, some as partly compositional. But given that there are so many, how are we to decide which ones are which? It is this task that I hope the methods described here will support.

1.3 Analysing their semantics

I want in this section to briefly describe a few relevant aspects of the approaches that have been taken to VPCs by other authors, focusing on analyses that seem to be widely cited, echoed or otherwise representative.

Bolinger, 1971(p.xi) describes VPCs as “an outpouring of lexical creativeness that surpasses anything else in our language”, and describes them almost exclusively in terms of their productivity and hence in terms of compositional or additive meaning. He observes that “The everyday inventor is not required to reach for elements such as roots and affixes that have no reality for him. It takes only a rough familiarity with other uses of head and off to make them available for head off”.

Bolinger’s strongest claim is that “In its core meaning (though not necessarily in figurative extension) the particle must contain two features, one of motion through location, the other of terminus through result”(p.85). He cites a number of examples in order to argue that we don’t see manner, time, place and stance adverbials in combining to form VPCs. He does, however, recognize that there are cases where neither of these features can be seen, and explains these exceptions in terms of aspect, acknowledging his debt to a number of earlier writers, (e.g. Kennedy, 1920; Fairclough, 1965; Live, 1965). He states that “the bulk of phrasal verbs whose meanings have deviated from

the more or less literal sum of its parts" can be explained in terms of "perfectivity" where perfectivity refers to the satisfactory performance or completion of an action (p.97). Compare, for example, the sentences *Madeline smashed the vase*, and *Madeline smashed up the vase*. The contribution of the particle *up* in the second seems to be to add the meaning that the act of smashing has been completed. The same contribution can be said to be made by a range of particles:

(5) Gussy drank down his orange juice.

(6) Bingo dragged out the last of his cocktail.

While later writers have offered a more fine-grained analysis of aspect (e.g. Bame, 1999), Bolinger warns that "one can easily indulge in aspect splitting (and get nowhere)" (p.101). Bolinger attempts to relate the spatial and the aspectual usage of the particle, claiming that "there is no real borderline between non-aspectual and aspectual uses of the particles, but rather a gradient. If a noun is described as in a condition resulting from an action, the nature of the condition will impute some kind of aspect to the action"(p.98).

Bolinger's description of a "semantic gradient from highly concrete meanings of direction and position to highly abstract meaning akin to aspects" (p.110) is very useful in analysing a large number of "phrasal verbs". However, it is is not difficult to come up with examples for which he fails to account. We cannot, for example, understand the VPCs found in the sentence *They made out on the couch* in terms of the verb *made* and the directional or perfective particle *out*.

That this is the case is discussed by Fraser, 1976. He draws a distinction between "systematic" and "figurative" VPCs. The former he defines as "those in which a consistent process of verb modification is occurring", and like Bolinger he recognises a straight forward distinction within this group between "combinations where the particle appears to have retained an adverbial force" and those in which the particle, rather than serving as an adverbial, appears to modify the meaning of the verb", although he doesn't explore the second group. Where he differs considerably from Bolinger is in his claim that "the systematic cases amount to only a small part of the total part of the total verb-particle combinations in the language". He identifies the dominant group as "those in which we have nothing but a frozen form".

While I believe this suggestion that the majority of VPCs are unanalysable to be wrong, my disagreement with Fraser concerns the frequency of items and I agree with him as to the existence of such a class. Unanalysable VPCs (at least synchronically)

seem to constitute a significant group. However, a number of writers have expressed doubts about the distinction of analysable from analysable VPCs. For example, Gries, 2000 argues that “the meaning of a verb phrase cannot always be categorised as being either fully idiomatic or totally literal - rather there are many cases where the meaning is somewhere between these two extremes.” And there is a substantial literature within the “cognitive grammar”(eg. Langacker, 1991) tradition which emphasizes this observation.

Lindner, 1983, for example, points that in previous work ”every analysis posits a group of VPCs for which the particle is not viewed as bearing any meaning of its own”. She picks Fraser out for particular criticism and claims that “particles almost invariably do code some part of the meaning of the VPC... In order to recognise the meaningfulness of the particle, however, it is necessary to recognise more than just the single “literal”, or concrete meaning that Frazer assumes...”. This criticism of Fraser seems to be a misreading. As I explained above he distinguishes between the adverbial and aspectual meaning, and he gives numerous examples where “systematic” VPCs have a non-”literal” meaning. Lindner’s objection seems to be to the sense-enumerative approach to word meaning, and she offers an interpretation of their meaning in term of “space grammar” (eg. Langacker, 1982). While I recognise the strong limitations of sense-enumeration here, a computational grammar must have some kind of finite set of possible word senses, albeit through sense enumeration or generative lexical rules, and there seem to be VPCs in which the semantics cannot be understood in terms of any such capacity.

A great deal more detailed analysis has been offered of the various phenomenon described above, but I will stop here. In summary, I think that we can take the following lessons. It seems from the literature that any grammar must be able at least to support cases where the following applies:

1. Both or either of the verb and the particle contributes its simplex meaning. For example, in the sentence *they took the boy back* both the verb and the particle seem to be contributing their standard semantics, in the sentence *they sought out the best deal* only the verb seems to be contributing its standard meaning, and in the sentence *they turned his brother in* only the particle would appear to be contributing its standard simplex meaning.

2. Both or either the verb or particle contributes a meaning which is systematic across idioms but has no corresponding simplex meaning. For example, in the

constructions *kick off* or *kick in*, the word *kick* would seem to mean ”start”, a sense which is shared by these VPCs but cannot be attributed to the simplex verb *start*. Examples of non-standard but systematic particle semantics were seen above in the discussion of the perfective usage *tear up, smash up*.

3. The construction cannot be said to be compositional in any way, and cannot be usefully analysed in terms of the meaning of any recognised meaning of either the verb or the particle. Examples that would seem to be in this category are *he was jacking up, he set her up*. This category also includes those items where there is a clear metaphorical derivation which is idiosyncratic and cannot be analysed without recourse to another ”domain” , e.g. *he had carved out a niche for himself, he wound her up all week*.

These are the key decisions that must be made about any VPC, and these (with some simplifications that I will detail below) are the decisions that will be the focus of the techniques I describe in this paper.

1.4 Defining the Current Task

1.4.1 Some Necessary Simplifications

Before defining the task that I am approaching in this project, I need to refer to an important principle. This is mentioned by Lipka, 1972. He writes that “we can only ascertain that a certain construction is different with regard to the cluster of semantic features which would normally be expected - i.e. is idiomatic - if we know which features are contained in the dictionary” (p.80). In order to recognise whether or not a given verb -particle is compositional in one of the ways listed above, we need to have a full analysis of the various ways in which different word forms can contribute to the meaning of constructions. This is something that we currently do not have, and the creation of which is beyond the scope of the project described in this paper. It is desirable then to simplify the task.

Given that we have a satisfactory list of the ways in which individual lexical items can behave outside of idiomatic constructions, it is reasonable to attempt to automate our analysis of whether a verb or particle is contributing its standard simplex meaning to a VPC. The problem is that there seem to be significant meanings for particular verbs and particles that are found only in VPCs, However, without a satisfactory list of

the those ways in which they can systematically contribute a sense which is exclusively found in VPCs we cannot satisfactorily judge whether this is the case, and particularly we cannot automate it. I will therefore simplify the task I am approaching in this paper in the following ways. I will attempt to produce a classifier which tells us whether each verb or particle in a VPC is contributing its standard simplex meaning. Each VPC can therefore be classified as falling into one of four classes:

1. Both the verb and the particle contribute their simplex meaning (e.g. *force out, take back*).

2. The verb but not the particle contribute its simplex meaning (e.g. *speak out ,buy up*) .

3. The particle but not the verb contribute its simplex meaning (e.g. *shell out, ward off*).

4. Neither the verb nor the particle contributes its simplex meaning (e.g. *hammer out, snap up*).

Where a particular verb or particle is judged not to contribute its simplex meaning, this could be because it has a meaning which is systematic but only within other multiword items or it could be because it is completely fossilized. We are not claiming to be able to distinguish this with current resources.

This simplification can be justified on two grounds. The first of these is that this reduced task, if it could be successfully performed, would be a very useful technique in and of itself, as it would allow us to identify those items which cannot be dealt with by the usual mechanisms of the grammar. Secondly, this distinction is a valuable step towards a full system because it enables us to isolate which instances need special consideration, and would aid us in observing greater systematicity.

Even given this simplification of the task, it is still not trivial for a human judge to decide whether any given verb or particle is contributing simplex meaning. The criteria we chose to use in deciding this are essentially whether the semantics of the simplex verb or particle can be used to helpfully decompose the construction. This comes down to a question of entailment, and whether we can say that the sentence involving the VPC entails certain statements involving the simplex verb or particle. If we can say that the statements involving the verb or the particle are entailed, then we can say that the item has standard semantics. The following are examples for each of the four cases:

1. If the VPC sentence is *Tom put the picture up*, does this entail that the picture has been put somewhere by Tom, and that as a consequence the picture is up? The answer seems to be yes, and we can classify the VPC here as fully compositional.

2. If the VPC sentence is *Richard finished up his paper*, then can we say that the paper is finished by Richard, or that as a result the paper is up? The answer to the first question is yes, but to the second is no. We can therefore say that the verb has standard semantics here, but the particle does not.

3. If the VPC sentence is *Philip gunned down the intruder*, can we say that the intruder has been gunned by Philip or that as a result the intruder is down? The answer to the former question would seem to be no, but to the latter question the answer is yes. We can therefore say that the particle but not the verb has standard semantics.

4. If the sentence is *Richard and Bethany made out*, then can we say that the two individuals involved made, or that they were out. The answer to both questions seems to be no, and we can therefore say that the verb is completely non-compositional.

While I must reiterate that the judgement is not complete, these criteria seem to allow us to make a motivated consistent and useful judgement.

I have been made aware of a very similar use of entailment in analysing the semantics of VPCs in Lohse et al (in preparation). They are interested in the factors that influence particle placement, and develop a semantic classification of VPCs which promises to be a significant factor. They classify the VPC in terms of the dependence or independence of the verb on/from the particle and vice versa. They say that in, for example *figure out* or *pick up*, the verb is dependent on the particle, whereas in *push down* or *lift up* it is independent. This is very similar to my compositional/non-compositional distinction, in that we might say that in a fully compositional VPC the verb and the particle have independent meanings whereas in non-compositional items their meanings are interdependent. They use entailment in making this judgement, so that if the VPC entails the verb or the particle that item (the verb or the particle) is said to be semantically independent.

1.5 About the Data

The aim of this project is to explore the usefulness of a number of features in automating the decision outlined above. In order to do this we needed to have a dataset against which to evaluate the decision, and to train an eventual classifier. Since there is no such ”gold standard” body of data we annotated our own. I mentioned above that the data used in this project was that generated by methods described in Baldwin and Villavicencio, 2002. I will briefly explain how this worked.

Baldwin and Villavicencio, 2002 describe the creation of a classifier for identifying VPCs from raw text. They attempt three techniques and then combine them into a final classifier. They develop and test this over the Wall Street Journal Section of the Penn TreeBank (M.P. Marcus and Marcinkiewicz, 1993). They evaluate by providing a precision score calculated over the combined total of 4,173 types extracted by the various methods described, and a recall score relative to a random sample of VPCs from the Alvey Natural Language Tools Grammar (Grover et al., 1993). 200 were taken from the grammar, but the score is relative to only the 62 that were found in the WSJ by a manual search.

1. They part-of-speech tag and lemmatise the WSJ data. They then locate particles by tag, and look back up to five words to the left for the head word of the VPC. They take a canonical set of 73 particles to filter the extracted items. This gives 135 VPCs, with precision of 1.000, recall of 0.177 and F-Score of 0.301 (see section 4.2.2 for an explanation of these evaluation metrics).

2. They chunk parse the WSJ. Again they look back up to five words and apply the additional stipulation that the only items that can occur between the verb and the particle are noun chunks, prepositional chunks adjoining noun chunks, and adverb chunks found in a closed set of particle pre-modifiers. Again the gold-standard particles were used to filter out items. The scores obtained were 0.786 for precision, 0.693 for recall, and 0.737 for the F-measure.

3. They again chunk parsed the WSJ, but this time in the hope of improving recall, they look for the 73 canonical particles in prepositional and adverbial as well as particle chunks. They take each verb chunk and look up to five words to the right for such a chunk. For each item that they obtain the analyse the following :

   (a) The chunks that occur between the verb and the “particle”

Valency	Precision	Recall	F-Measure
TRANSITIVE	0.836	0.800	0.817
INTRANSITIVE	0.842	0.711	0.771

Figure 1.1: Performance Scores for Valency Specific VP Extraction

(b) The chunks that occur immediately after the particle/preposition/adverb chunk to check for a clause boundary or NP complement

(c) The clause context of the verb chunk for extraposition of NP complement.

These are used to test for consistency with the properties of VPCs as described at the beginning of this chapter. In many cases where an NP occurred either after the particle, it was unclear whether it should be analysed as a VPC or as a prepositional verb or a free verb preposition combination, and in those where it was before the particle it was unclear whether the "particle" was the head of PP postmodifying an NP. In an attempt to resolve this ambiguities using the frequencies of verb-preposition, preposition-noun and verb-noun bigrams over the chunker output. These tests are used to generate frequency-based features and combined in various ways to classify the item. The best combination achieves precision of 0.695, recall of 0.871, and an F-measure of 0.773.

While the third method improves recall considerably, it does so with a substantial loss of precision. The researchers therefore tried to benefit from the varying strengths of the different methods by combining their outputs into a single classifier. Having embellished the classifier with seven new lexical and frequency based features (the frequency of the particle in the corpus, the frequency of deverbal noun and adjective forms of the VPC in the corpus, the verb lemma, the particle lemma and the number of letters in the verb lemma) it achieves its best scores of 0.889 for precision. 0.903 for recall and 0.893 for the F-measure.

The data that I used was slightly different from that finally achieved by these methods. Rather than the output of a single classifier, I was given the results of two classifiers, one for transitive, and one for intransitive verbs. The motivation for this was to allow me to make use of these judgements in obtaining my features. The price of this additional information was a reduction in the quality of performance, as can be seen in Figure 1.1.

1.5.1 Annotation

This section describes my approach to annotating this data. The most important decision I made was to annotate by type rather than by token. This raises the significant problem of polysemy. As is the case with any lexical item, VPCs can have more than one sense. When I attempt to describe all instances of a particular VPC in a corpus with one judgement, I am oversimplifying enormously, since the likelihood is that it will encompass items which have a different sense and for which the judgement simply isn't true. I had to make various simplifications in order to get around this.

In annotating any given VPC, I looked closely at the data found in this particular corpus. If one dominant sense was observed, with a minimum ratio of 4:1, then I classified it accordingly. Any less than this and I simply discarded it from the set. My approach can be justified on a number of grounds. Firstly the individual annotation of 5250 instances by type would have not have been possible for the researcher to achieve in a limited time-frame. Secondly, and perhaps most significantly, any unseen data that I might attempt to classify using these techniques are not going to be annotated or split up according to sense. And thirdly, I can defend the discarding of polysemous items on the grounds that I want to explore the usefulness of the features in classifying any lexical item, and as long as it is understood that the results produced are not fully indicative of the kind of results that might be obtained on unseen data, this is the best available way to accurately gauge this.

Another problem that I came up against is that of item frequency. Of the 843 VPC types that are present in the data, a total of 263 of these occur more than four times. Having less than four instances of an item is not only worrying from the point of view of developing useful features, it is hardly enough to confidently classify an item. I therefore rejected any item that occurred fewer than four times. This left a total of 263 VPCs. Of these there were 25 items that I took to definitely not be verb particles (using the criteria described above) in the data, 28 items that were prohibitively polysemous, and 30 items that while they were acceptable types, had instance data that was far too noisy to use, and therefore had to be discarded. This left 180 VPC types over 2034 tokens. I took this to be enough data for this preliminary study. The final data can be found in Appendix A.

1.6 Future Chapters

The rest of the paper will focus on describing the implementation of two main features, and their evaluation. It will be structured as follows.

- Chapter two will describe a feature that employs a measure of the "substitutability" of the verb and particle for any given item. This substitutability measure concerns the extent to which the verb or particle of any given VPC may be replaced with a verb or particle of a similar semantic class to form other VPCs that are attested in the data. The intuition here is that if it reflects systematic patterns in this way then it is more likely that the verb or particle concerned have their simplex meaning.

- Chapter three describes the implementation and evaluation of a measure of degree of semantic relatedness between the VPC and its component verb. The intuition here is that if a VPC is semantically close to the verb then it is more likely that the verb contributes its simplex meaning.

- Chapter four describes the combination of these two features in the training and evaluation of a decision tree classifier.

- Chapter five reviews the findings, offers some discussion of the performance of the features, and points to some directions for future research.

Chapter 2

Substitutability and Compositionality

2.1 Justifying the Approach

In much of the literature on VPC semantics that I briefly surveyed in the last chapter, questions of compositionality are closely associated with the issues of word-formation and productivity. So what is the reason for this association? Productivity is any pattern of combination of linguistic units (be they words or smaller morphemic units) that can be used to produce novel forms. The number of VPCs in the language continues to grow, most often according to recognisable patterns of formation, and many already established items in the lexicon can be accounted for in terms of active patterns of combination/productive lexical rules. However, as with other kinds of lexical items there are many VPCs that cannot be accounted for in this way. Such items we refer to as lexicalised. Huddleston and Pullum, 2002 explain this term as follows:

> "The converse of productivity is lexicalisation: words that are or were earlier morphologically analysable but which could not be formed with their present meaning by the current rules of word formation are said to have been lexicalised. The implication of the term is that properties of these words have to do with specified individually in the dictionary rather than being consistent with the grammatical rules of word-formation."(p.1629)

These definitions of productivity and lexicalisation can be directly related to what I have been saying about compositionality. For reasons explained in the last chapter we can say that any VPC whose meaning cannot be found by combining verb and particle meanings found in a lexicon (which we have said for the purposes of this investigation must be restricted to simplex meanings) should be treated as non-compositional. Clearly any VPC that can be accounted for in terms of a productive process of combination of such items, is compositional in some way according to this definition. While

it is not necessarily the case that a lexicalised item is non-compositional, it would certainly appear to be more likely that it is so. Given this close association, might productivity be a useful predictor of compositionality?

This Chapter describes a technique, adapted from existing work on collocation discovery, that might be used in predicting whether any given VPC is productive, and indeed whether a particular constituent of a VPC is contributing its simplex meaning. I should stress that neither the measure of productivity that it employs, nor its application to the task of predicting compositionality, are proposed as entirely satisfactory linguistic tests, but rather that they might be useful indicators that could, when combined with other features, be useful in buiding the classifier that I am aiming at in this paper.

In order to find out whether a word can be explained in terms of a productive process of word-formation we need to see to what extent any item can be explained in terms of a rule that accounts for other items found in the data. Looking across the VPCs in the set that I am using in this paper, a tendency can be observed for verbs with similar meanings to occur with particles of similar meanings. For example, among those found in the Wall-Street Journal Corpus we find that the particle *up* can be combined with a set of verbs that seem to concern a person's movement of her/himself from one place to another by means of a mode of transport (e.g. *drive up, ride up*, etc, while the verb *move* can be combined with a number of particles concerning spatial direction (e.g. *move over, move across, move away, move out, move in, move back, move off*, etc). And if we take either of these examples we can see that it can be generalised to make a rule. If we have another verb that similarly denotes self-propelled movement, such as *cycle* or *pogo* then we can combine it with up to form acceptable strings *cycle up, pogo up* . Similarly if we take another directional particle such as *aside* or *through* then this can likewise be combined with the verb *move* to form the acceptable strings *move aside* and *move through*. These tendencies would seem to constitute productive processes since they can be used to produce new items, through principled combination. We want then to find a way to discover whether for any given VPC we can find other combinations where verbs of the same class have been combined with this particle or particles of a similar class, and vice versa. In order to do this we are going to need to find a way of obtaining for each item a list of verbs and particles of the same class. Fortunately, as I will show, we have access to many of the resources we require.

2.2 Relevant Work

While empirical work has been done on measuring the productivity of particular lexical rules (e.g (Briscoe and Copestake, 1999)) and even on those rules that concerning multi-word items (Copestake and Lascarides, 1997; Copestake, 2001), no work has been done directly on discovering these rules or patterns. However there is a body of work which has grown up around a different but apparently related problem that might be of use to us. The idea that we can distinguish, for any language, between word combinations that can be entirely accounted for by a generalizable syntactic or semantic rules, and those that cannot, has long been discussed (if until recently little utilised) in work on collocation. While collocation is a different phenomena from lexicalisation, the concern with distinguishing idiosyncratic combination is the same. (Manning and Schutze, 1999) write that ”A collocation is any turn of phrase or accepted usage where somehow the whole is perceived to have an existence beyond the sum of the parts” (p.29). Might the techniques employed for recognising collocations be of use to us?

2.2.1 Collocation and Compositionality

I want here to briefly describe some work on collocation extraction that I think provides us with some useful techniques for the task of identifying productivity. Pearce (2001a, 2001b, 2002) proposes a method for recognising collocations by automatically performing a substitution test. He extracted bigrams from about 5.3 million words of the British National Corpus. For each pair the aim was to provide a score, where a high score meant a strong collocation. The assumption is made that any phrase is one lexical realisation of a concept. Other lexical realisations are produced by taking each word in the phrase, and performing a substitution using words from each appropriate WordNet synset (see 2.3.1 for more information) for that word. Two probabilities are then found. The first is the joint probability of n such independent trials, one trial from each synset, thus giving the likelihood that any realisation of the “concept” will occur. The second probability is the frequency count of a given phrase in the data, normalised by the sum of all the others. The difference between these then gives the “deviation” of the probability of the phrase occurring from that given an assumption of free substitutability. Pearce points out that the method assumes that the word sense is known. Thus for the shared task in Pearce, 2002, when used on data which is not tagged for word sense, the method can be expected to suffer from inclusion of synsets relevant containing the wordform but not the relevant word sense. For evaluation of

that task, the output was compared to 4,152 entries occurring in the data that occurred in a dictionary. No numbers are given for the task, only graphs, but the results seem to compare favorably with more traditional approaches to collocation extraction.

In attempting to detect whether a particular bigram is a collocation or not Pearce is looking at whether there is evidence that the bigram is more than a free combination of words. When this is done for a kind of bigram that represents a variety of multi-word item, what he is essentially doing is testing to see if there exist lexical rule/s that can be used to account for the word, or if a collocation tendency needs to be posited to account for it, which in the case of a multi-word item might mean lexicalisation. This is essentially the same thing as we wish to do here.

Pearce is very much aware of the relationship between collocation and non-compositionality. Indeed in his 2002 paper he presents his approach as follows:

> This supervised technique is based on the assumption that if a phrase is semantically compositional, it is possible to substitute component words within it for synonyms without a change in meaning. If a phrase does not permit such substitutions then it is a collocation. More specifically to the experiment described in this paper, the *degree* to which such changes are possible is obtained and this is used as a score.

While this suggestion of a direct opposition between compositional phrases and collocations might be misleading since there are many examples of collocations that are compositional, he is assuming as we are that substitutability is a clue to compositionality. At least one other attempt has been made to relate collocation to issues of compositionality. (Lin, 1999) describes his aim as one of automatically identifying "non-compositional expressions" by statistical methods. As we will see, its approach is very similar to that of the previous researcher. It justifies this approach as follows:

> "The intuitive idea behind the method is 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."

The method was to extract dependency relations from a corpus parsed with Minipar (Lin, 1998b), and use this information to extract collocations, by taking a dependency relationship as a collocation if its log-likelihood ratio is above a certain threshhold. For a set of these collocations Lin then substituted each of the words with a word with a similar meaning. The list of similar meanings was obtained by taking the 10 most similar words according to a corpus-derived thesaurus, the construction of which is described in Lin, 1998a. A frequency count and mutual information value was then found for each item produced by this substitution. The mutual information of two

points x and y, is a relationship between the probability of observing them together, and the probability of observing them independently:

$$I(x,y) = log\frac{P(x,y)}{P(x)P(y)}$$

The mutual information value of a collocation triple was found by taking a collocation to consist of three events - the type of dependency relationship (A), the head lexical item (B), and the modifier(C). The mutual information then is the logarithm of the ratio between the probability of the collocation (where the probability space is all possible collocation triples) and the probability of the head, type and modifier occurring together, taking the head and modifier to be independent given the type:

$$I(A,B,C) = log\frac{P(A,B,C)}{P(B|A)P(C|A)P(A)}$$

Lin's initial observation, given these scores, is that "a phrase is probably non-compositional if such substitutions are not found in the collocation database, or their mutual information values are significantly different from that of the phrase" (p.319). Having refined his algorithm Lin provides some examples that suggest he has identified a successful measure of compositionality. He offers an evaluation where an item is said to be non-compositional if it occurs in a dictionary of idioms. This produces the unconvincing scores of 15.7% for precision and 13.7% for recall.

2.2.2 Some problems with this approach

I mentioned before that lexicalisation is different from collocation, and we need to bear in mind where this difference lies, and where it might cause us problems. The biggest problem, with Lin's work, and with my use of the approach here, as I see it, is that there is no way to distinguish between compositional collocations and non-compositional or lexicalised forms. There are frequently exceptions to any productive rule. It is a flawed but useful technique.

2.2.3 Application to VPCs

In making use of these techniques to detect non-compositional VPCs it is necessary to note the difference between VPCs and the kind of items that the work above was concerned with. The most notable difference is that we are dealing with a specifically defined group of items. This has a considerable effect on both the frequency of the

items, and the number of candidate items which might be considered when looking for substitution, and in turn would seem to effect the kinds of statistical measures that are open to us. This distinction is also relevant in the choice of the kinds of lexical resources that are available, particularly when it comes to substituting particles.

2.3 Lexical Resources

While Lin uses a corpus-derived thesaurus, I have access to no such resource, and to create one would take considerable time. For this reason I decided to focus on utilising what ready-made resources I could get my hands on in substituting verbs, and hand-crafting my own resources for particles. The next three sections will describe the resources I am using.

2.3.1 WordNet

Miller et al., 1990 describe WordNet as “a dictionary based on psycholinguistic principles”. It consists of a substantial part of the English lexicon split into groups of words that have a similar meaning (known as ”synsets”) that are organised into a hierarchy employing multiple inheritance. As such it is usually regarded as having more in common with a thesaurus. For the work described here I used WordNet 1.7.1. This contains a total of 195,817 unique word-sense pairs arranged into 111,223 different sets of synonyms. It consists of a separate database for each of nouns, verbs, adjectives and adverbs, each of which is organised differently. For example nouns are organised into sets of synonyms that are hierarchically arranged according to hyponymy, adjectives and adverbs are organised according to an *n*-dimensional space in bipolar clusters of antonyms, and as we will see verbs are organised into synsets according to entailment relations. A psycholinguistic motivation is claimed for these differing approaches to each wordclass, details of which are given in Miller et al., 1990.

WordNet contains over 24,169 verb word-sense pairs, in 13,214 synsets. These are split into 15 different groups. Fourteen of these are said to refer to different semantic domains (e.g. Consumption, Motion, Perception) and concern events or actions. The other group (stative verbs) does not have domains in common, but rather is connected by the fact all contained verbs refer refer to states. These classes are then subdivided further, sometimes from more than one root node. For example motion verbs are split into more than 500 synsets, and are derived from one of two roots, either move, make

a movement, as in to move in one position or location, or move, travel as in movement from one position or place to another.

As I mentioned above, the verbs in WordNet are organised according to a different principle to that used for nouns. They employ various kinds of entailment in defining relations, the principal types of which I describe here. Miller et al., 1990 state that there are four kinds of entailment, which can be divided into those that are temporally inclusive of one another and those which are not. Of the former group, one important relation is defined a troponymy. A troponym v^1 of a particular verb v^2 can be distinguished by the statement "to v^1 is to v^2 in a particular manner". For example we can say that "to stroll is to walk in a particular manner" or "to doze is to sleep in a particular manner". The other temporally inclusive entailment relation is that shown by word pairs like snore-sleep, swallow-eat. The first verb can be said to entail the second verb, but they are not necessarily temporally co-extensive, as eating and sleeping involve elements other than swallowing and snoring. Those relations where the verbs are not temporally inclusive of one another are again split into two. One is backward presupposition, so that to say that one has arrived presupposes that one was travelling. The other is cause, where the verbs may be described as causative and resultative such as with the pairs give-have, learn-know.

The most significant thing for our purpose is that the verb hierarchy tends to have many fewer levels than that for nouns, and that they tend to cluster on particular levels rather than spread equally. This is significant here as it means that some synsets are likely to be considerably larger than others. One other thing to note is that the verb hierarchy contains many short phrases as well as individual words, which I deal with by simply rejecting all but simplex items.

2.3.2 Resources for Particles

While we have a substantial resource for verbs, we are not so spoilt for particles. WordNet includes adverbs but there are various problems here. The principal issue is that there is a very large number of adverbs that never occur as particles. While the fact that many adverbs do not occur in the WSJ data might arguably suggest a significant limit on productivity, these limits are not restricted to VPCs and thus would not give an indication of productivity for this particular class of complex lexical items. I might constrain my use of WordNet to only consider those items that are seen in the data. However given the small number of items that need to be considered it seems that I

might more usefully produce my own classification and resources.

While Baldwin and Villavicencio, 2002 consider 73 possible particles while searching for relevant constructions in the WSJ, there are only 27 different particles found in those extracted.

up	down	off	out
over	in	back	ahead
to	abroad	on	about
together	around	through	along
across	away	by	forward
since	without	astray	throughout
behind	aside	under	

Once I'd disregarded those items that I regarded to be wrongly classified as a VPC there were only 11 different particles attested:

ahead	around	away	back
down	in	off	out
over	through	up	

It is unclear which of the three groups should be used for performing substitutions. I have opted for producing a classification of all particles, as it is easy then to simply reduce the range of items being used if it is found to give better results. This distinction between the approaches will not affect the kind of items that are produced through substitution, but it will be significant when I come to give a score for productivity and need a score for how many items could be produced through substitution but were not attested.

My classification divides the possible particles into three (overlapping) classes — those concerning temporal position, those concerning spatial position and those concerning spatial direction. The classes are as follows:
Temporal position:

after	afterward	afterwards	before
during	since	beforehand	throughout
at	past		

Spatial-direction

```
across  along  around  away
back  backward  backwards  down
forth  forward  from  hither
onto  through  thru  to
toward  towards  up  via
aside  into  about  against
ahead  astray  at  beyond
in  off  on  out
over  past
```

Spatial-position

```
about  above  abroad  against
ahead  among  astray  behind
below  beside  between  beyond
by  facing  in  near
nearby  off  on  out
over  past  thereabouts  throughout
under  upon  upstairs  within
without  at  with  across
```

A number of items are included in more than one group (e.g. *at* is in both the temporal and the spatial position group and *past* is in all three). One interesting thing to note is that the particles that are attested seem only to be those concerning spatial position and direction, which might be a useful observation when we come to constrain substitution. It is useful to sort the spatial words into two categories as while, for example, *in* would seem to be available as both a direction and a location, a word like *back* can only refer to a direction and not a position.

2.4 Some Example Output

Using the above resources, I was able to perform substitutions to generate possible items. Take for example the VPC *force up*. The particle *up* belongs to the ”spatial-direction” set of particle as described above. Substituting from the set produces the following items:

```
force_down      force_over      force_aside     force_into
```

force_along	force_across	force_at	force_toward
force_to	force_backward	force_from	force_towards
force_in	force_back	force_about	force_off
force_ahead	force_beyond	force_onto	force_around
force_thru	force_forth	force_away	force_on
force_hither	force_against	force_out	force_backwards
force_astray	force_forward	force_through	force_via

The verb *force* has 9 different meanings in WordNet/belongs to 9 different WordNet synsets. The synsets, with the hypernyms sets displayed for the purpose of disambiguation, are as follows:

{coerce, pressure,force} (Hypernym set => {compel, oblige, obligate})
{impel, force} (Hypernym set => {cause, do, make})
{push, force} (Hypernym set => {move, displace})
{force, thrust} (Hypernym set => {compel, oblige, obligate})
{wedge, squeeze, force} (Hypernym set => {move, displace})
{force, run, drive, ram} (Hypernym set => {thrust}
{pull, draw, force} (Hypernym set => {move, displace})
{force} (Hypernym set => {act, move})
{storm, force} (Hypernym set => {penetrate})

Substituting from these sets produces the following VPCs:

coerce_up	hale_up	pressure_up	impel_up
push_up	thrust_up	wedge_up	squeeze_up
drive_up	ram_up	pull_up	draw_up
storm_up			

Of all these only the following substitutions are found in the corpus data:

force_down	force_aside	force_out	push_up
drive_up	pull_up	draw_up	

2.5 Putting it to Work

Once I had found a way of generating this data, I needed to find a way to turn it into a useful score. The simplest method would simply be to count the number of attested

items that are produced by performing substitutions for any item, and use that as its score. This score, however, would not be particularly useful, as it would not take into consideration the fact that the number of synonyms that items have will vary greatly. A higher score would then be given to a word with 25 synonyms which gave rise to two attested items than to an item with only one synonym where this one possible substitution was attested. This does not seem to give a good indication of relative productivity. I needed rather to find a score that gives an indication of how the number of attested items compares to the number of possible items. For this I found a score based on the ratio between the number of attested items and the number of possible items:

$$SubstitutionScore = \frac{Numberof AttestedItems}{Numberof PossibleItems}$$

This use of a ratio between possible and attested items as a way of measuring the degree of rule productivity is suggested in (Briscoe and Copestake, 1999), and is adapted for multi-word items (specifically compound nominals) in (Copestake, 2001). There is one significant problem remaining. Items are arranged according to sense. For this reason there are going to be items taken into account that are simply a result of polysemy. Since different words will belong to varying numbers of different synsets, this will provide considerable noise. This is the same problem as Pearce (2002) mentions, and there seems to be no easy way around it. One solution might be to only accept items that occur with a particular frequency or above, and indeed some effort has been made to provide information about the frequency of the words in WordNet using the "semcor" data. However, this provides very limited information as the frequencies are acquired from a very small corpus. Indeed when I tried blocking all items under a certain very low frequency threshold using the semcor frequencies I found this did little to distinguish between slightly less common and rare or idiomatic words.

2.6 Results

We want to discover how useful the substitution score is in predicting the semantics of the VPC. It is therefore interesting to look at the distribution of the scores across the compositionality classes. The mean number of attested substitutions for items in which the verb is judged to be contributing a standard meaning in the gold standard data is 2.0147, while the mean score for those in which the verb is judged to not be contributing such a meaning is 1.6518. This seems in line with the hypothesis, but when I conducted a t-test to compare the distributions of scores in these two groups,

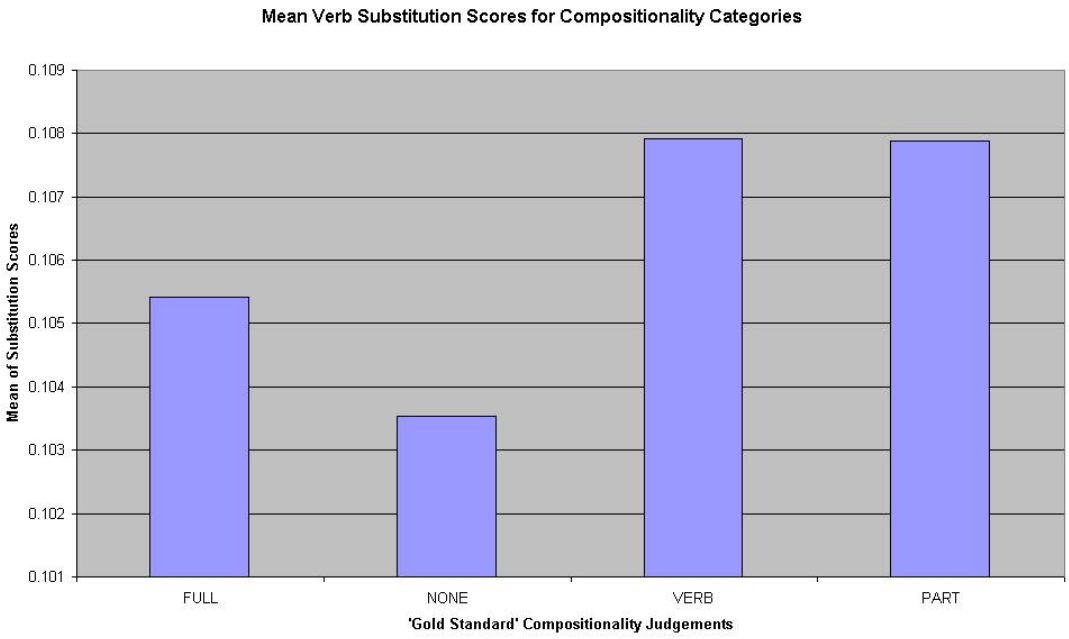

Figure 2.1:

I obtained an insignificant result ($t = 0.94687$ $df=178$ $p > 0.05$). Using the ratio-based substitution score described above, the mean scores are 0.10655 and 0.10447 respectively for these same categories. A t-test comparison of these two categories gave an insignificant result($t = 0.10957$ $df=178$ $p > 0.05$). The mean verb substitution scores (ratio-based) over the four categories given by the compositionality judgements in the gold standard data can be seen in figure 2.1.

The results for particle substitution were considerably more encouraging. The mean number of attested substitutions in which the particle is judged to be contributing its standard meaning is 6.59 while in those where the particle is judged to not be contributing in this way, it is 3.31. A t-test comparing these two groups gives a highly significant score ($t= 4.9236$ $df=178$ $p < 0.01$). Using the ratio-based score, the mean for those where the particle is judged to be compositional is 0.15 and for those where the particle is not compositional it is 0.08. A t-test again gives a very significant score ($t=4.8708$ $df=178$ $p <0.01$). These scores clearly support the hypothesis that the more substitutable a particle is, the more likely it is to be contributing a standard meaning. When only those particles that occur in attested VPCs are made use of in the substitutions, the mean substitution scores obtained for those items where the particle is judged to be compositional is 0.45 and for those where it is not the mean score is 0.246. Figure

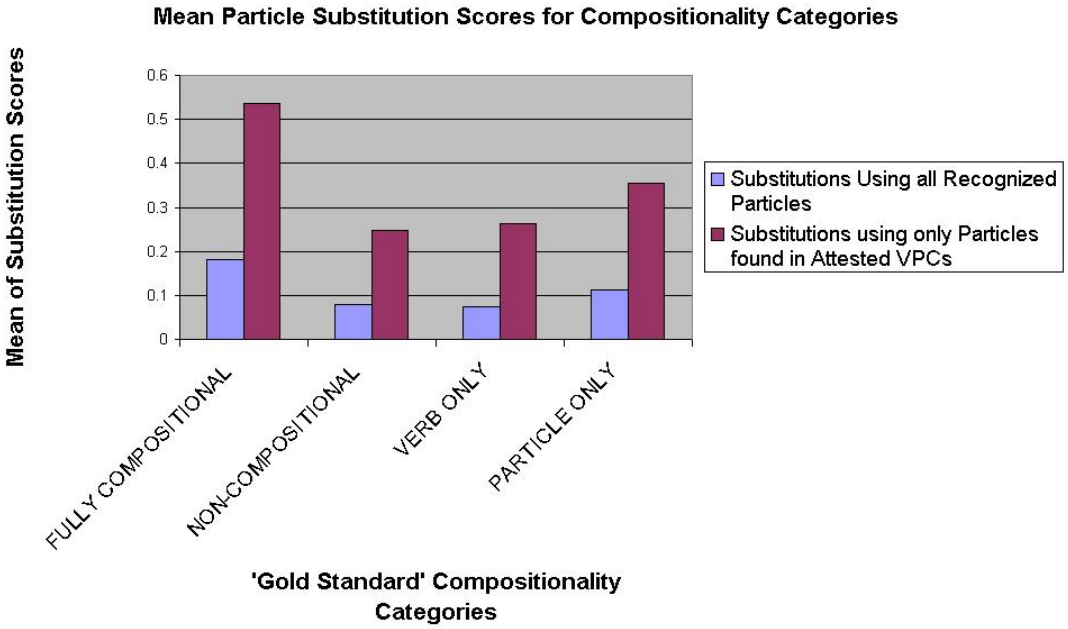

Figure 2.2:

2.2 shows the mean substitution scores across the four compositionality classes.

Chapter 3

Semantic Relatedness and Compositionality

This chapter will describe the implementation of another feature that promises to be useful in inferring the degree and nature of compositionality of a VPC. The intuition behind the approach is that identifying the degree to which a VPC is semantically related to the simplex verb might be a useful feature in distinguishing a compositional from a non-compositional VPC i.e. if a VPC can be shown to be semantically similar to a simplex sense of the verb form that it contains, then this could be a good indicator that the construction is compositional.

3.1 Automatic Measures of Semantic Similarity

An initial distinction can be made between two approaches to the measurement of semantic relatedness that have been shown to be effective. One is to use existing lexical resources such as the WordNet hierarchy that we saw in the last chapter. The other is to use the contexts of the words. The former methodology makes the assumption that since WordNet is a network of semantic relations it should be possible to use distribution in the network as a measure of semantic relatedness. This work is by no means to be rejected out of hand, and as we will see later it is indirectly useful here, but we have the simple problem that VPCs are simply not dealt with in a satisfactory way by any of the resources that are available to use (see Villavicencio and Copestake, 2002 for discussion of the patchy VPC coverage of different lexical resources). While some VPCs are included in WordNet, only a small percentage of the items that we are interested in are accounted for. It is therefore on the context of words that we must

focus.

3.2 Semantic Relatedness and Context

Corpus-based work on the automatic discovery of semantic relatedness is based on the idea that words that have a similar meaning are likely to occur in a similar context. This idea has received empirical support from, for example, Miller and Charles, 1991. They took the judgements of human subjects as to the similarity of a set of pairs of nouns, and compared this with a measure of the similarity between the contexts of each of the nouns. They found that the degree of similarity of the contexts of two words could be used to predict the human similarity judgements. Work on using this observation to make automatic judgements might be usefully split into two types of approach — those that use context to mean simply those words that surround the item, and those that try to use syntactic (dependency) information to zero in on the contextual information that we might expect to be most distinctive to the word in question. A full review of the literature on this would take a considerable amount of space, and is not justified here. I will instead briefly describe the most widely known work.

Perhaps the most often cited works using the former method are those of Deerwester et al., 1990 and Schutze, 1992. Deerwester et al., 1990 use whole documents as contexts and build a matrix where rows represents words and columns represent documents. The frequency of occurrence of each word in each document is then entered in the matrix. Singular value decomposition is then used to reduce dimensions, to arrange the space according to the important associative patterns. Schutze, 1992 uses a similar technique, but takes the context to be windows of 1000 characters rather than whole documents.

Of the later kind of work, the most comprehensive account of a successful study is provided in Grefenstette, 1994. Grefenstette, 1994 uses syntactic context to measure the similarity between words. He works with a number of datasets and begins by parsing them and extracting the grammatical relations for each word. He focuses exclusively on the grammatical relationship between nouns, adjectives and verbs. Verbs, which is what we are interested in here can be related in his work to subjects, direct objects and indirect objects. The grammatical relation is said to be an attribute, so that for the verb “gave” in the sentence *the man gave a book to the boy* the attributes of *gave* are *man-SUBJ, book-DOBJ,* and *boy-IOBJ*. The similarity between two items is then found using a weighted Jaccard Similarity Measure. Each attribute is given a

global weight between 0 and 1 for informativeness, based on how many often it appears and how many other items it appears with. The similarity between two words is then measured by taking the sum of the scores of the attributes that are shared by the comparison items divided by the sum of the scores of the attributes that are unique to the comparison items. While both these approaches have been shown to be very successful on particular kinds of data, the nature of the data I am using here means that I cannot hope to directly apply either of these methods and achieve success. The crucial issue is word frequency. The grammatical relations method relies upon the exact match of attributes, and as such it can only produce useful results where there are a considerable number of instances. The document co-occurence technique would seem to overcome this since its multidimensional measure of association does not rely on exact lexical co-occurence. It does however rely upon distinctive contextual associations that still only emerge over a number of occurrences. For this reason I propose a hybrid method, which zeros in on the informative items by extracting the grammatical relations for a word, but rather than looking for the exact match between lexical items, a measure of the semantic similarity is used between the set of items that occur as grammatical relations of the VPC, and those that occur as relation of the simplex verb. Thus by using the intuition that a verb's arguments (be that verb simplex or complex) are indicative of its meaning, we are able to make use of semantic similarity measurements in finding the distance between the more frequently occurring simplex items that occur in the argument slots. It also means that I was able to make use of ready-made resources. This technique could make use of semantic space models, but since none were readily available, and it would have been time-consuming to build them to the required standard, I have focused on using WordNet for this preliminary study.

3.3 Obtaining Grammatical Relations

First I split the corpus into two sets — that which contains the VPCs and that which doesn't. I then parsed all the data using the RASP parser (Briscoe and Carroll, 2002). This parser provides detailed grammatical relations in its output, as seen below.

```
("Pierre" "Vinken" "," "61" "years" "old" "," "will" "join" "the" "board" "as" "
a" "nonexecutive" "director" "Nov." "29") 1 ; (-26.653)
(|ncsubj| |join:9_VV0| |Vinken:2_NP1| _)
(|dobj| |join:9_VV0| |board:11_NN1| _)
```

```
(|obj2| |join:9_VV0| |Nov.:16_NPM1|)
(|ncmod| _ |Vinken:2_NP1| |Pierre:1_&FW|)
(|ncmod| _ |year+s:5_NNT2| |61:4_MC|)
(|ncmod| _ |year+s:5_NNT2| |old:6_JJ|)
(|ncmod| _ |Vinken:2_NP1| |year+s:5_NNT2|)
(|ncmod| _ |director:15_NN1| |nonexecutive:14_JJ|)
(|detmod| _ |director:15_NN1| |a:13_AT1|)
(|ncmod| |as:12_II| |board:11_NN1| |director:15_NN1|)
(|detmod| _ |board:11_NN1| |the:10_AT|)
(|ncmod| _ |Nov.:16_NPM1| |29:17_MC|)
(|aux| _ |join:9_VV0| |will:8_VM|)
```

For my VPC data, because RASP is not always able to detect VPCs, I simply treated the relations that the parser marked as being those of the simplex verb as being those of the VPC. I was then able to simply extract the significant relations from this. I was interested in those items taken in subject and direct object positions. While the output on the whole was very useful, it of course had problems in dealing with the VPCs. The most consistent error it made was treating the direct object of the VPC as an indirect object with the particle as the head of a prepositional phrase.

While this results in a reduction of useful attributes and a problem with valency judgements, it didn't result in any mistaken items occurring in our extraction. There is an additional problem with simply comparing argument slots of any verbs. This is that the arguments of some verbs can be syntactically expressed in more that one way. This phenomena is commonly referred to as diathesis alternation. With VPCs we find a considerable number of instances where the verb seems to be contributing similar selection preferences in the VPC but where the arguments occur in different positions. This can be attributed to alternation. The variety of alternation that causes particular problems with VPCs is that referred to as causative-inchoative alternation (e.g. Levin, 1993, pp 27–30). There is both a transitive and an intransitive form for the verbs, and in the intransitive form the direct object of the verb is expressed in the subject position. With VPCs it is frequently the case that a VPC has an intransitive form, which displays this alternation. For example we say that *he heated the soup* but not that *the soup heated*, while we can say that *the soup heated up*. I did not implement any successful technique for dealing with this here.

3.4 Measuring semantic distance between arguments

Having extracted these grammatical relations, I then wanted to find how similar the set of relations for each VPC was to those for the simplex verb. My first approach was to take each lexical item that occurs in subject and direct object position for each VPC and compare it in turn with each lexical item that occurs in the same position for the simplex verb. I then took the mean (and later the median) of the score for each position to be the distance of that set. I used the following technique to measure the semantic distance for each pair of items.

3.4.1 Using WordNet

Budanitsky and Hirst, 2001 evaluated the performance of five different methods that have been proposed for measuring the semantic distance between words in the WordNet Hierarchy. These have all been implemented and freely distributed by Pedersen, 2002. I tried all of the methods implemented, and found that proposed by Jiang and Conrath, 1997 to perform best.

Jiang and Conrath, 1997 combine WordNet with corpus statistics. The basic idea is that semantic distance should be reflected in the sum of the edges that make up the shortest path that links the nodes at which the two words occur. They observe, however, that because the density of nodes varies in different parts of the hierarchy, just using the straightforward sum of the edges does not give an adequate measure of semantic distance. In other words it is a mistake to treat all edges as representing the same semantic distance. We want therefore to weight each edge so that its score more accurately reflects semantic distance. In order to do this we calculate the information content of each node in a corpus. We then say that the strength of a link is the difference between the information content value of the child concept and its parent. We factor in the depth of the node in the hierarchy, the local density of nodes, and the type of connection, to produce a weight for all edge. The distance between two nodes then is the sum of these weights along the shortest path linking them.

3.5 Backing off to Word Classes

While treating the sets of arguments item by item provided me with promising results, it has a number of problems that I subsequently tried to remedy. Most significantly using the mean score leaves the score very vulnerable to the minority of the items

that occur through parser errors while the median does not seem to be discriminating enough.

The problem of wanting to describe the set of items taken in particular argument slots by verbs is not a new one. A considerable number of methods have been proposed for finding the best way to make use of what they tell us about the verb. What I am doing is essentially trying to compare the preference of each VPC and verb for taking a particular kind of item in a particular argument slot. It seemed that a reasonable way to progress then was to employ the notion of selectional restrictions. Selectional restriction refers to a verb's constraints on the classes of items that can occur in particular argument positions. They have traditionally been expressed as hard constraints (and often referred to as selectional constraints), but Clark, 2001 models the preferences of verbs in probabilistic terms. In this context we more accurately refer to selectional preferences, and I will prefer that term in this paper. A considerable amount of work has been done on automatically acquiring the selectional preferences of given verbs. Again, a lot of this has employed the WordNet hierarchy, and focused on trying to find the class in WordNet that best describes the set of arguments. It is this that I next attempted.

3.5.1 Extracting Selectional Preferences

In order to find the class that best described our sets of arguments, I used Diana McCarthy's slightly modified implementation of the method proposed by Li and Abe, 1998. McCarthy, 2001 makes use of the WordNet hierarchy to produce a "treecut" or a set of classes across the hierarchy which dominates all the nodes at which the words taken in a particular argument position are located, and to give a preference score to each class in the cut. The first step of this method is to populate the hierarchy with information about the frequency of occurrence of particular classes in the data. This is initially done by assigning for each item a score to each class of which the item is a direct member (i.e. immediate hyponym). This score is the sum of the frequencies of all the items that occurred at that class, divided by the total number of classes of which the item is a direct member. These scores are then propogated up the hierarchy so that the probability at the root is 1.

The next step is to use this populated hierarchy to find a measure of preference for the classes. The probability of each class is found using Maximum Likelihood

estimation:

$$p(c) = \frac{freq(c)}{N}$$

where c is the class and N is the sum of frequencies for all classes. The measure of preference employed by McCarthy, as suggested by Li and Abe, 1996 is the probability of the class given the verb divided by the probability of the class. This is referred to as an "association score". The final step is to obtain the treecut. A cut is chosen using the principle of minimum description length, where the best treecut model is said to be the one that has the shortest description length; the description length is the sum of the model description length and the data description length.

3.5.2 Making use of Selectional Preferences

The output of this method is, for each VPC and simplex verb, a set of WordNet classes each of which has been given a score indicating the verb's preference for taking words of that class in each argument position. How then can we use this to compare across verbs? I will look at a few possible methods. First I must make a point about the results that I obtained. Presumably because of our sparcity of data, the cuts taken were always from the nine top level classes. While these are useful they greatly restrict how I might use the information. There are few methods that are intuitively appealing but proved to be of little use. Those that I tried are as follows:

1. The most straightforward method is a binary test as to whether the class which is most preferred by the VPC in each position is the same as that most preferred by the simplex verb in the same position. This gives a useful indication for a handful of items, but it also predictably gives a large number of false negatives.

2. Another slightly more flexible binary method is to look for the VPCs preferred class in the set of classes in the treecut of preferred classes for the simplex. The problem with this is that given the fact that the classes always choose from the nine top level classes, this is likely to give us very many false positives.

3. Had I obtained classes lower down the hierarchy, I might have considered using a distance measure over the hierarchy as explained above in order to tell the distance between the most preferred class in each position for the VPC and those for the simplex verb. However, given that all the preferred classes are top-level, the semantic distance measure is not going to be a useful comparison.

4. The method of comparing favourite classes directly causes us to miss out on a number of similar items. The most preferred class can be employed in a slightly different way by taking the most preferred class of the VPC and seeing what score is obtained by the simplex verb on this same class. Given the small set of classes from which the treecuts are taken, this allows me to obtain a score for most pairs. Given the way in which this is derived, it can be treated as a probability that the items taken in a particular argument position by the simplex will be of the VPCs most preferred class.

I tried out all four methods, and found the fourth to be the most successful.

3.6 Evaluation

There are two main kinds of evaluation that need to be carried out on the methods described in this chapter. Firstly I need to evaluate the performance of the technique proposed for the subtask of measuring semantic similarity. I then need to evaluate how useful this feature performs on the overall task of predicting compositionality. I will also offer an empirical test of the assumption that underlies this chapter.

3.6.1 Materials for Evaluation

The evaluation of techniques for automatically detecting semantic relations is notoriously difficult (see for example the discussion in (Grefenstette, 1994)). I chose the popular method of comparing the judgements of the system with human judgements. In order to generate this data, I presented 9 human volunteers with a set of 80 verb particle and particle pairs, and asked them to rate how similar in meaning they thought them on a scale of 0 to 10, with 10 being very similar and 0 being not similar at all. The volunteers all had English as their first language, none of them had any formal training in Linguistics or Computer Science, and none were informed of how the data was to be used until after they had completed their judgements. Of the 80 pairs, 40 were taken from my annotated set of VPCs, the VPC being paired with its simplex verb. I took 10 from each category (i.e. 10 where it was fully compositional, 10 where only the verb contributes etc.). For the remaining 40, 20 were chosen from the set of VPCs and defining simplex verb pairs found as an appendix to (Fraser, 1976), with 10 of these being examples were the verb was that included in the VPC. The remaining 20 were randomly selected combinations. The aim with this sample was to situate the data in

the context of a set that was fairly evenly distributed in terms of type and degree of similarity. Once the score were all in, I obtained a mean for each pair, and this mean was used for my evaluation.

Before explaining how I made use of this data, there are a few observations that should be made about it. The items were isolated, and the volunteers were not given any information about what word class any of the strings belonged to. It is interesting to note that where the simplex verb in the VPC was denominal, the participants judged the VPC to be similar to the simplex item, whether or not it could occur in simplex form. For example *gun down* was judged to be highly similar to *gun*, and *jack up* was judged to be highly similar to *jack*, even though neither would appear to occur as a simplex verb. While any full technique would need to acknowledge the compositionality of such items, for reasons explained in Chapter One, mine currently does not, and the method I am using to measure semantic similarity is clearly unable to handle them appropriately.

3.6.2 Testing the Value of the Feature

Before going on to evaluate the module, I want to briefly offer an empirical test of the premise of this chapter. If, as is assumed, a VPC that is more semantically related to the simplex verb is more likely to be compositional, then I should be able to test this using the human judged data. I split the human-judged VPCs into two groups - those in which the verb was labelled as contributing a standard meaning in the gold-standard data and those in which was not. I then performed a t-test comparing the distribution of the human-judged similarity scores across the two samples. The outcome was highly significant (t=3.2857 df=24 $p < 0.01$). I also performed a linear regression to test the correlation between the human-judged similarity scores and the compositionality judgements from the gold-standard data. I did this by assigning a score of 1 to each of the VPCs where the verb was labelled as contributing standard semantics, and a 0 where it was not. The outcome of the regression was again highly significant (r= 0.55700, F(1,178) = 10.796, p = 0.0031189). These outcomes clearly support the hypothesis that the more semantically related a VPC is to it component simplex verb the more likely it is to be at least partly compositional.

An interesting effect can be observed by looking at the distribution of the scores over the four compositionality classes, as shown in figure 3.1. In line with the above tests we can see that those VPCs that are completely non-compositionality according

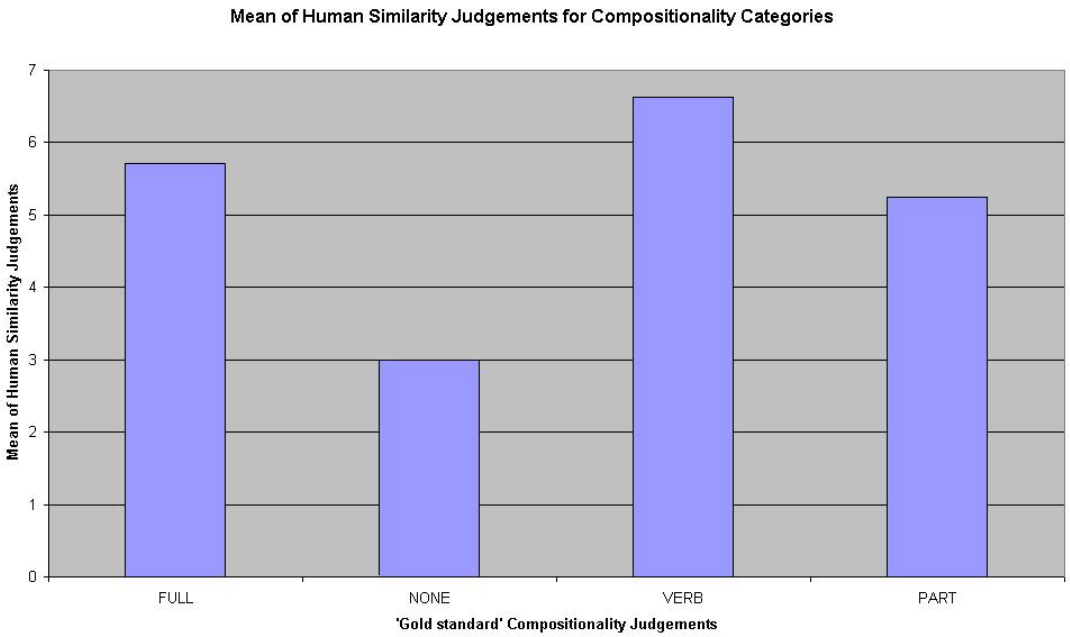

Figure 3.1:

to our criteria have a significantly smaller mean similarity score than that in which the verb is judged to contribute standard meaning (FULL and VERB). More surprisingly, those in which the particle alone is judged to be compositional also scored quite highly. This might perhaps be accounted for by the fact that looking over our data we can see that VPCs in which the verb is de-nominalised such as the two quoted above tend to have a compositional particle.

3.6.3 Evaluation of Module in Isolation

The evaluation of my technique's performance at evaluating the semantic similarity is quite straightforward given the data. I need to look at what correlation there is between the similarity judgement of the computer with that of the human participants. In order to do this I performed linear regression on the data. I did this first for the technique of taking the mean distance scores of the lexical arguments, and began with the items in subject position. This showed an insignificant relationship ($r = .12431$, $F(1,13) = .20405$, $p = .6589$). My first test on the similarity scores for the items in the direct object position was again insignificant ($r = .55510$, $F(1,18) = .00809$, $p = .9293$). However, when I looked at the distribution of scores I found that there was one extreme outlying instance that had been given a similarity score more than twice

that of any other. This was *buy back* whose sole direct object item in our data - *share* - also occurred 57 times as the direct object of the simplex verb *buy*. The extremely high score seems to have been caused by the fact that the argument of this sparse VPC found such a high number of exact matches amongst the verb arguments. When I removed this I found a highly significant relationship (r = .55510, F(1,17) = 7.57117, p = 0.0136). For the preferred class measure, a very nearly significant correlation was found for the subject (r = .42502, F(1,19) = 4.1886, p = 0.0548) and no significance for the direct object position (r = .54395, F(1,8) = 3.36177, p= .1041).

3.6.4 Evaluation of techniques performance on main task

We next wanted to look at how the scores promise to perform on the main task of semantic classification. The distribution of the scores for the lexical arguments based score over the compositionality classes can be seen in figure 3.2, and those for the preferred class measure can be seen in figure 3.3. All of those show the tendency we have anticipated, with the fully compositional and verb-compositional scores being markedly higher for the similarity measures that they are for the non-compositional and particle-compositional classes. Whether this tendency is strong enough to be useful we will see when we examine their performance in the classifier. It is useful to note that the direct object slot appears to be more useful than the subject slot in predicting semantics. This is in line with research in word-sense disambiguation which has found the direct object slot to be more useful in discriminating word senses (McCarthy et al., 2001).

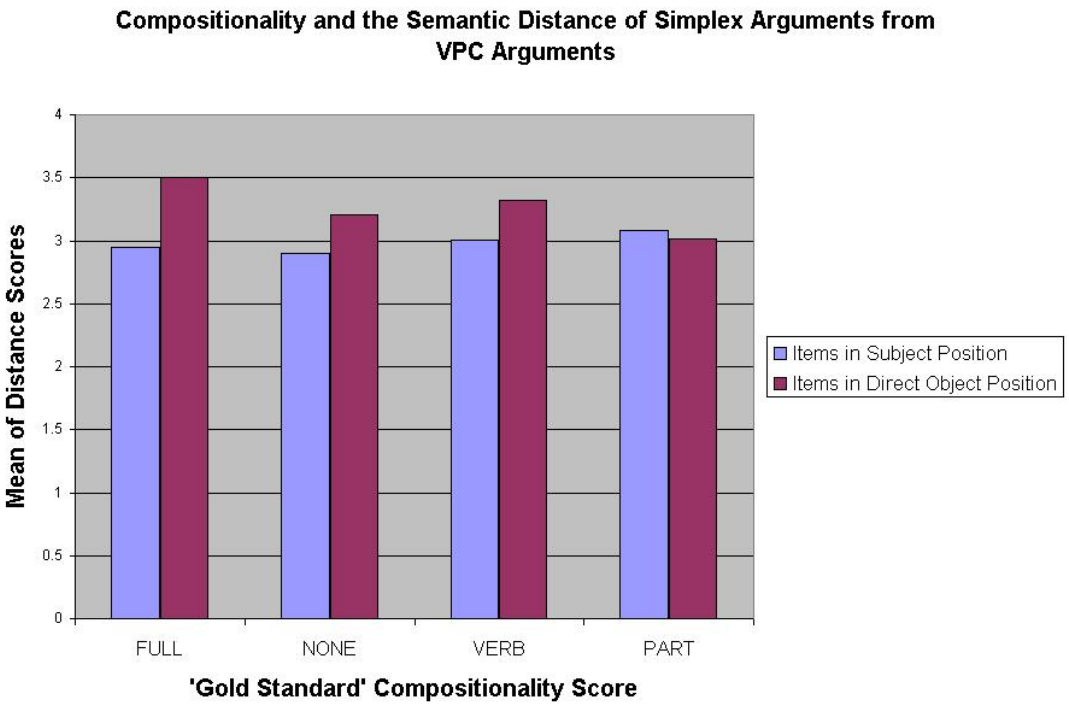

Figure 3.2:

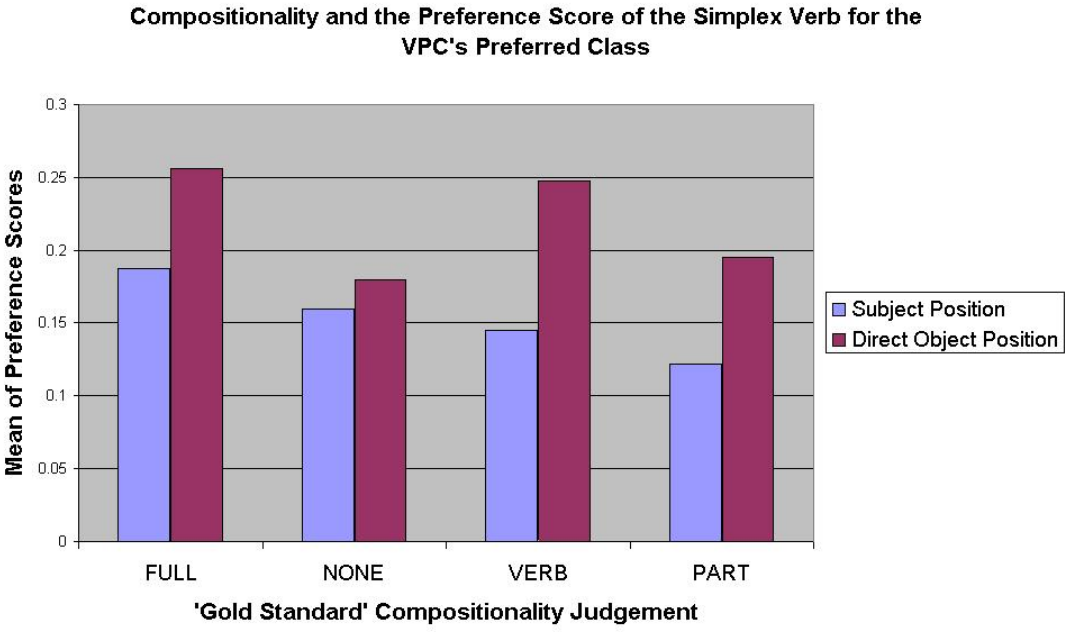

Figure 3.3:

Chapter 4

Building a Classifier

This Chapter will describe the creation and evaluation of a Classifier that makes use of the features described in the last two chapters. Although I have offered some initial results for the individual features, I want to be able to observe how useful they are in actually making predictions and how well they perform in combination.

4.1 One last feature

In creating my classifier there is one last feature of which I tried to make use. Villavicencio and Copestake, 2002 describe an examination of the VPCs that are found in a set of five paper and electronic dictionaries and lexicons. They point out that a number of items that they have identified are not found in any of the dictionaries or lexicons and they judge that the majority of these are at least semi-compositional. They note that this is probably partly intentional on the part of the compilers of the works, and that one of the dictionaries they are working with, the Collins Cobuild Dictionary of Phrasal Verbs (Moon, 2000), explicitly states that the literal meanings and combinations are not always given. I decided to add the existence or non-existence of the items in the dictionaries as a feature on the grounds that it should be a good predictor of whether an item was compositional or not. I was provided with details of all the VPCs that were found in these works. I then compared this list to my list of items. If I found that an item was in our list but not found in the dictionaries, then I gave it a score of 1, otherwise the items were given a score of zero. The distribution of these items across the classes can be seen in 4.1.

	Not in Dictionary	Total Number of Items	Mean Score
Fully Compositional	5	37	0.135
Non-compositional	4	88	0.45
Verb only	0	31	0.45
Particle Only	2	24	0.83

4.2 About the Classifier

I built my classifier using the weka machine learning toolkit (Witten and Frank, 2000). This implements a number of different algorithms for analysing data and creating classifiers. While I tried out a number of these on my data in the process of creating a classifier, the point here is to provide a preliminary study of how the features perform, and so I will describe only the classifier that I found to produce the best results. This was the C4.5 decision tree learner. I will briefly describe how this works.

4.2.1 About C4.5

A decision tree is a hierarchically arranged series of decisions, where the exemplars are partitioned at each node depending on the value of a particular feature. A decision is made at each node of the tree depending upon the values of different features. The unknown instance follows a route down the tree according to the value that it has for these features. It will eventually reach a leaf and this will give it the predicted class. The aim in creating such a tree is to find the series of branching decisions that best uses the selection of features available to give the best classification of unseen data. Branching decisions can be implemented for both nominal and numeric values. In order to work with the latter we need to discretise the values of the feature. This is usually done by creating a binary split, with the dividing line being halfway between the highest and lowest scores.

The principal that is employed in the creation of decision trees, is that we want to create the daughter nodes with the greatest degree of purity, i.e. the least amount of variance of class. We want to find the feature that is most successful at creating these, which is the one that is best at predicting the class, and we do this by working out which features gives us the most information for the set of instances. To do this the algorithm uses the information gain measure. This score tell us how much information is gained by the addition of a feature. It is calculated as the difference in entropy between the situations with and without the knowledge of the feature. The entropy(H), of a set of

class labels C is written as:

$$H(C) = -\sum_{c \in C} P(c) log_2 P(c)$$

where P(c) is the probability of the class label estimated from relative frequencies in the training set.

This measure allows selection of the attribute that gains us the most information to split on. The procedure is repeated at each node (on each subset) until the data cannot be split any further. Ideally this would be because all of the members of the set are of the same class, but often it is not as clean as that. Some post-pruning is done on these trees, based on the estimation of error for the training data. Pruning is done by replacing subtrees by single leaves, or by raising subtrees to subsuming nodes.

4.2.2 How the Scores are obtained

In order to evaluate the performance of the classifier, it needs to be run over a set of data which is disjoint from the training set. One common method is to "hold back" a certain percentage (typically 20%) of one's data for testing and train on the remainder. However, this is far from ideal here as it will result in a loss of a part of the already very limited training data. I decided therefore to employ the technique of ten-fold cross-validation. The data is randomly divided into ten parts. Each of these parts is then held out in turn and the remaining nine-tenths used for training the classifier, and performance scores taken for its performance on the tenth part. The ten score are then averaged to give overall performance scores. This technique is often preferred because it minimizes the effects of bias and variance in the data. The scores that I give are precision, recall and the F-measure. These are based on the relationship between the number of true positive, false positive and false negative judgements. The first two are found as follows:

$$precision = \frac{truepositives}{truepositives + falsepositives}$$

$$recall = \frac{truepositives}{truepositives + falsenegatives}$$

The F-measure is found by combining these two scores in the following way:

$$F = \frac{2 \times recall \times precision}{recall + precision}$$

These scores are widely preferred in statistical NLP to simple error counts because they reward true positives. I use them here because they are widely employed and

thus form a basis for comparison. I will also give accuracy scores which refer to the percentage of all items that were correctly classified.

4.2.3 The Terms of Classification

Based on the analysis that I described in Chapter One, my annotated data supports the description of items according to the following four classes.

1. VPC is fully compositional, entailing both the verb and the particle.
2. VPC is completely non-compositional, entailing neither the verb or the particle.
3. VPC entails the verb, but not the particle.
4. VPC entails the particle, but not the verb.

I showed some initial significance scores for the various features in the preceding chapters, and when I came to build a classifier I found that the performance of the various techniques was in line with my expectations. In a nutshell, what I found was that the following scores were the best for each category:

- For verb substitution I found that the best results were obtained where the score was the number of attested items formed by verb substitutions from WordNet, divided by the total number of items that could be created by verb substitution.
- For particle substitution, the best score was given by substituting from the categories that I created from all particles, rather than just those that are attested in the data. Again I used the attested items divided by possible items score.
- For similarity judgements, the most valuable score seemed to be given using the mean of the distance scores for each argument position, that was obtained for each VPC and simplex pair using the WordNet based distance score.

I will only describe the results obtained using the best features.

4.3 Results

The results that were obtained need to be put in context. Of the 180 VPCs that I have annotated, 37 are classified as fully compositional, 31 are classified as having only a

```
psubscore <= 0.0625
|   psubscore <= 0.03125: NONE (64.0/20.0)
|   psubscore > 0.03125
|   |   psubscore <= 0.032787
|   |   |   SubjectSimilarity <= 2.690894: NONE (3.5/1.5)
|   |   |   SubjectSimilarity > 2.690894: VB (3.5/0.5)
|   |   psubscore > 0.032787
|   |   |   SubjectSimilarity <= 2.627573: VB (6.43/1.71)
|   |   |   SubjectSimilarity > 2.627573: NONE (2.57/0.57)
psubscore > 0.0625
|   SubjectSimilarity <= 2.893561
|   |   DirectObjectSimilarity <= 3.107612: NONE (25.83/12.67)
|   |   DirectObjectSimilarity > 3.107612
|   |   |   psubscore <= 0.081967: NONE (3.74/0.37)
|   |   |   psubscore > 0.081967: FULL (11.54/2.52)
|   SubjectSimilarity > 2.893561
|   |   psubscore <= 0.065574: PART (2.59/0.59)
|   |   psubscore > 0.065574
|   |   |   SubjectSimilarity <= 3.232197
|   |   |   |   SubjectSimilarity <= 3.147862
|   |   |   |   |   DirectObjectSimilarity <= 3.42759
|   |   |   |   |   |   psubscore <= 0.125: NONE (5.88/2.0)
|   |   |   |   |   |   psubscore > 0.125: FULL (10.31/5.16)
|   |   |   |   |   DirectObjectSimilarity > 3.42759: VB (4.78/0.56)
|   |   |   |   SubjectSimilarity > 3.147862: PART (5.52/0.52)
|   |   |   SubjectSimilarity > 3.232197: NONE (29.81/16.25)
```

Figure 4.1: Decision Tree for Four Class Classification Task using Substitution and Similarity Feature

verb that contributes standard semantics, 24 as having only a particle that contributes its standard meaning, while 88 are classified as completely non-compositional. This distribution of items means that on the four class classification we are able to classify 48.88% of the items correctly just by calling them all completely non-compositional. This score might then be taken as a baseline. However, in terms of the present task it is not a particularly useful one. The point of the task is that we would like to be able to describe the semantics of VPCs in terms of the simplex items in the lexicon, partly because we wish to aim at maximum decomposition and partly for reasons of economy. Therefore, although treating all items as completely idiosyncratic gives us a reasonably high overall accuracy, since it is precisely the approach to classification we are trying to avoid, it would in fact give be of no value to us.

When trained on both the Substitution and the Similarity features using the four class labels, the classifier produced the decision tree that can be seen in Figure 4.1. It is useful to look at the detail of this. The first split is according to whether the particle substitution score is greater than or less than/equal to 0.625. If it is the former then the next rule asks whether the similarity score for the subject position is greater than or less than/equal to 2.893561. If it is less than this figure than the next rule asks whether the Direct Object Score is greater than or less than/equal to 3.107612. If it is less than/equal to this figure then it is said to be completely non-compositional, while if it is higher than this another question is asked that says if the particle substitution score is less than/equal to a certain figure it is non-compositional but if it is higher than it is fully compositional. While these splits are not completely in line with the predictions I would have hoped our figures would make, none of them run completely counter to expectation. Elsewhere in the tree, however there are branches that are more surprising. There are, for example two rules that state that if the Subject Similarity score is below a certain figure then it is compositional, but is above then it is not. This is the opposite of what I would like to be able to predict, but it is very deep in the tree and seems to be due to overfitting rather that the features being dysfunctional.

The performance of this tree in classifying over the tenfold cross-validation is shown in figure 4.2. As we might expect there is a pretty impressive performance in the identification of non-compositional items, and variously less impressive rates on the other classes. The overall Accuracy for this classifier across all classes is 45%. When I add the dictionary-based feature that was described at the beginning of this chapter, the scores seen in figure 4.3 are obtained. While there is an improvement in both precision and recall for verbs there is an overall fall in performance, and the

Class	True Positive Rate	Precision	Recall	F-Measure
NONE	0.739	0.516	0.739	0.607
FULL	0.216	0.296	0.216	0.25
VERB	0.194	0.3	0.194	0.235
PART	0.083	0.286	0.083	0.129

Figure 4.2: Scores for Four Class Classification Using Substitution and Similarity Features

Class	True Positive Rate	Precision	Recall	F-Measure
NONE	0.727	0.5	0.727	0.593
FULL	0.162	0.261	0.162	0.2
VERB	0.226	0.333	0.226	0.269
PART	0.042	0.125	0.042	0.063

Figure 4.3: Scores for Four Class Classification Using Substitution, Similarity and Dictionary Features

accuracy is 43.33%.

Chapter 5

Conclusion

This paper has suggested, and offered some initial testing of, some features for predicting the semantics of an unseen verb-particle construction (referred to throughout as VPCs) from corpus data. The motivation for this is that we would like to be able to predict whether any such construction might be analysed using simplex lexical entries, or whether it has a non-decomposable semantics and thus requires a idiosyncratic entry. I devised a four class classification scheme. This is described in detail in chapter one, but to recount, the four classes were as follows:

1. Both the verb and the particle contribute their simplex meaning (e.g. *force out, take back*).

2. The verb but not the particle contribute its simplex meaning (e.g. *speak out ,buy up*) .

3. The particle but not the verb contribute its simplex meaning (e.g. *shell out, ward off*).

4. Neither the verb nor the particle contributes its simplex meaning (e.g. *hammer out, snap up*).

A set of 180 VPCs were annotated according to this scheme, and this was used as a gold-standard set for first testing the intuitions underlying the features and then training and testing a classifier.

The first feature I looked at was the extent to which the verb or particle of any given VPC may be replaced with a verb or particle of a similar semantic class to form other VPCs that are attested in the data. The intuition here is that if it reflects systematic patterns in this way then it is more likely that the verb or particle concerned have

their simplex meaning. I performed these substitutions using the synsets of WordNet for verb substitution, and my own semantic classes for particle substitution. A score was assigned to each item based on the ratio of items produced through substitution to those produced that were actually found in our data. An initial examination of the distribution of the substitution scores over classes formed from the compositionality judgements supported the intuition, and when a t-test was performed I found the particle substitution scores to be very significantly higher (t=4.8708 df=178 $p < 0.01$) for those VPCs where the particle was judged to be contributing its simplex meaning as compared with those where it was judged not.

My second feature was a measurement of the degree of semantic relatedness between the VPC and its component verb. The intuition here is that if a VPC is semantically close to the verb then it is more likely that the verb contributes its simplex meaning. To support evaluation, I obtained the judgement of 9 human volunteers as to the semantic similarity of 80 pairs of VPCs and 80 VPCs and simplex verbs, 40 pairs of which were VPCs found in our data paired with their constituent simplex verbs. I then evaluated the hypothesis by splitting these 40 VPCs into two groups - those in which the verb was labelled as contributing a standard meaning in the gold-standard data and those in which it was not - and performing a t-test comparing the distribution of the human-judged similarity scores across the two sample . The outcome was highly significant (t=3.2857 df=24 $p < 0.01$). The next step then was to automatically measure the semantic distance of VPCs from their constituent simplex verbs. In order to do this I compared their selectional preferences. My first approach was to take each lexical item that occurs in subject and direct object position for each VPC and compare it in turn with each lexical item that occurs in the same position for the simplex verb. I did this using existing techniques for measuring semantic distance between words using the WordNet hierarchy (Jiang and Conrath, 1997). I then took the mean of the score for each position and used this as a measure of semantic distance between the VPC and the simplex verb. I next tried backing off from the lexical items to the class that best describes the set of arguments for each position of the VPC, using a existing technique (Li and Abe, 1998; McCarthy, 2001), and then using the same technique to find the probability that this class describes the arguments taken in each position by the simplex verb. I evaluated the performance of these techniques by comparing their similarity scores with those given by the human subjects. I performed linear regression for each and found a significant correlation using the first technique on direct object arguments (r = .55510,

F(1,17) = 7.57117, p = 0.0136), and a marginally significant score employing the second technique on subject arguments (r = .42502, F(1,19) = 4.1886, p = 0.0548).

As a final measure the two features were combined to produce a decision tree classifier. The best performing classifier using these features classified 45% of items correctly, obtaining f-scores of 0.607 on completely non-compositional items, 0.25 on fully compositional items, 0.235 on items where only the verb contributes a simplex meaning, and 0.129 where only the particle does so.

5.1 Discussion of Features

5.1.1 Verb Substitution

This is perhaps the most disappointing feature. The resources used, while not without problems, offered substantial coverage, and had been used on similar tasks in the past. We therefore might have depended on this to surpass the other features. However, its performance was the weakest overall. We might offer a straightforward explanation for this. As I pointed out earlier, WordNet is arranged according to word sense, and for most NLP tasks, we do not have any way of distinguishing between the various senses for any word form. This is exacerbated by the fact that the it contains idiomatic meanings along with the rest of the lexicon. It is difficult to see how improvements could be made with WordNet. There is potential for significant advances if we could have access to better sense frequency information, but populating WordNet with sense frequencies rather than wordform frequencies would require a prohibitively large collection of word sense annotated data.The key to a more successful substitution based approach is likely to be the employment of a corpus derived thesaurus. This would allow us access to considerable sense specific frequency information, albeit rather noisy.

5.1.2 Particle Substitution

This was in certain respects the most encouraging feature. The approach was supported by a t-test that found the particle substitution scores to be very significantly higher (t=4.8708 df=178 $p < 0.01$) for those VPCs where the particle was judged to be contributing its simplex meaning as compared with those where it was judged not. And while the overall classification of items as having only a particle that is compositional (the class to which we would expect the particle substitution score to most usefuly contribute) achieved the worst scores of all the classes, we should note that

it was by far the most difficult class to correctly classify. Firstly it is by far the least frequen class. Secondly and most importantly this class suffers significantly from the simplification we made in our experimental design. The semantics of particles is the aspect of VPC semantics that is least understood, and ignoring those items that contribute a systematic meaning that is not its simplex meaning, means that a large part of the semantic function of particles is ignored. Finally the particle substitution feature is working alone as while there are two features that we might expect to help in predicting the compositionality or otherwise of the verb, there are no other clues as to the status of the particle.

The most important thing to note here is that substitution does appear to be a useful predictor, and the fact that this was most effectively shown using a resource created specifically for the task suggested that the poor performance of verb substitution might be legitimately attributed to WordNet being ill-suited to the task.

5.1.3 Semantic Similarity

There is reason for some optimism here. The results obtained using the human similarity judgements offer a strong support for the usefulness of semantic similarity as a feature. What the experiments failed to show was whether we can obtain automatic similarity judgements that are good enough to enable its application. The performance here when compared with the human judgements was pretty poor, although we might expect a certain amount of disagreement with the human data, because of our consciously ignoring the compositionality of denominal or deadjectival items. However the techniques that are used were only heavily compromised versions of techniques that have been shown to be useful for measuring semantic distance. If a more substantial body of data was employed then we would be able to employ the proven techniques for measuring semantic similarity that I discussed in chapter 3. We might then reasonably expect to see a substantial improvement in the measurement of semantic similarity. One other important step that needs to be made is the investigation of techniques for testing the semantic distance of particles from VPCs.

Appendix A

Gold Standard Data

TRANS - Transitive
INTRANS - Intransitive
FULLY COMP - Entails both Verb and Particle.
VERB ONLY - Entails the Verb
PART ONLY - Entails the Particle

	TRANS	INTRANS	FULLY COMP	VERB ONLY	PART ONLY	
head_up,	1,	0,	0,	1,	0,	;
head_off,	1,	0,	0,	0,	1,	;
blurt_out,	1,	0,	1,	1,	1,	;
heat_up,	0,	1,	0,	1,	0,	;
hammer_out,	1,	0,	0,	0,	0,	;
shore_up,	1,	0,	0,	0,	0,	;
force_out,	1,	0,	1,	1,	1,	;
speak_out,	0,	1,	0,	1,	0,	;
hand_down,	1,	0,	0,	0,	0,	;
spread_out,	1,	1,	1,	1,	0,	;
fend_off,	0,	1,	0,	0,	1,	;
jack_up,	1,	0,	0,	0,	1,	;
sum_up,	1,	1,	0,	0,	0,	;
inch_up,	0,	1,	0,	0,	1,	;
drum_up,	0,	1,	0,	0,	0,	;
speed_up,	1,	1,	0,	0,	0,	;

```
bounce_back,    0,      1,      0,              0,              0,      ;
pay_back,       1,      0,      1,              1,              1,      ;
pay_off,        1,      0,      0,              1,              0,      ;
pay_out,        1,      1,      0,,             1,,             0,      ;
clear_up,       1,      0,      0,,             0,,             0,      ;
look_back,      0,      1,      1,              1,              1,      ;
take_up,        1,      0,      0,,             1,,             0,       ;
take_over,      1,      0,      0,,             1,,             0,       ;
take_off,       0,      1,      0,,             0,,             0,       ;
take_back,      1,      0,      1,,             1,,             1,       ;
seek_out,       1,      0,      0,,             1,,             0,       ;
prop_up,        1,      0,      0,,             0,,             0,       ;
slow_down,      1,      1,      0,,             0,,             0,       ;
pile_up,        1,      1,      0,,             0,,             0,       ;
team_up,        0,      1,      0,,             0,,             0,       ;
lose_out,       0,      1,      0,,             0,,             0,       ;
build_up,       0,      1,      0,,             0,,             0,       ;
fill_out,       1,      0,      0,,             0,,             0,       ;
ride_out,       1,      0,      0,,             0,,             0,       ;
work_up,        0,      1,      0,,             0,,             0,       ;
work_out,       1,      1,      0,,             0,,             0,       ;
turn_up,        1,      1,      0,,             0,,             0,       ;
turn_out,       1,      0,      0,,             0,,             1,       ;
turn_in,        1,      0,      0,,             0,,             1,       ;
turn_down,      1,      0,      0,,             0,,             0,       ;
turn_off,       1,      0,      0,,             0,,             1,       ;
turn_around,    0,      1,      0,,             1,,             0,       ;
find_out,       1,      0,      0,,             1,,             0,       ;
spin_off,       1,      0,      0,,             0,,             0,       ;
sell_off,       1,      0,      0,,             1,,             0,      ;
sell_out,       0,      1,      0,,             1,,             0,      ;
level_off,      0,      1,      0,,             0,,             0,      ;
end_up,         1,      1,      0,,             0,,             0,      ;
edge_up,        0,      1,      0,,             0,,             0,      ;
back_up,        0,      1,      0,,             0,,             0,      ;
```

```
back_off,       0,      1,      0,,             0,,             1,      ;
send_out,       1,      0,      1,,             1,,             1,      ;
grow_up,        0,      1,      0,,             1,,             0,      ;
dole_out,       1,      0,      0,,             0,,             1,      ;
follow_up,      1,      1,      0,,             1,,             0,      ;
firm_up,        1,      0,      0,,             0,,             0,      ;
stay_on,        0,      1,      0,,             1,,             0,      ;
cut_back,       0,      1,      0,,             1,,             0,      ;
cut_through,    1,      0,      0,,             0,,             0,      ;
catch_up,       0,      1,      0,,             0,,             0,      ;
catch_on,       0,      1,      0,,             0,,             0,      ;
shape_up,       0,      1,      0,,             0,,             0,      ;
move_up,        0,      1,      1,,             1,,             1,      ;
move_in,        0,      1,      0,,             1,,             1,      ;
move_out,       0,      1,      1,,             1,,             1,      ;
move_through,   1,      0,      0,,             0,,             0,      ;
pop_up,         0,      1,      0,,             0,,             0,      ;
step_up,        0,      1,      0,,             0,,             0,      ;
step_down,      0,      1,      0,,             0,,             0,      ;
step_in,        0,      1,      0,,             0,,             1,      ;
sign_up,        1,      0,      0,,             0,,             0,      ;
write_down,     1,      0,      0,,             0,,             0,      ;
write_off,      1,      0,      0,,             0,,             0,      ;
keep_on,        0,      1,      0,,             1,,             0,      ;
bail_out,       1,      0,      0,,             0,,             0,      ;
drive_down,     1,      0,      0,,             0,,             1,      ;
tie_up,         1,      0,      0,,             0,,             0,      ;
wake_up,        0,      1,      0,,             1,,             0,      ;
put_up,         1,      0,      0,,             0,,             0,      ;
put_in,         1,      0,      0,,             0,,             0,      ;
stir_up,        1,      0,      0,,             0,,             0,      ;
stand_up,       0,      1,      1,,             1,,             1,      ;
walk_away,      0,      1,      1,,             1,,             1,      ;
walk_out,       0,      1,      1,,             1,,             1,      ;
walk_in,        0,      1,      1,,             1,,             1,      ;
```

```
walk_around,    0,      1,      1,,             1,,             1,      ;
spring_up,      0,      1,      0,,             0,,             0,      ;
shell_out,      1,      0,      0,,             0,,             1,      ;
give_up,        0,      1,      0,,             0,,             0,      ;
give_out,       1,      0,      1,,             1,,             1,      ;
make_up,        1,      0,      0,,             0,,             0,      ;
carry_out,      1,      0,      0,,             0,,             0,      ;
watch_out,      0,      1,      0,,             1,,             0,      ;
bottom_out,     0,      1,      0,,             0,,             0,      ;
beef_up,        1,      0,      0,,             0,,             0,      ;
scoop_up,       1,      1,      0,,             0,,             0,      ;
start_up,       1,      0,      0,,             1,,             0,      ;
fight_back,     0,      1,      0,,             1,,             0,      ;
run_up,         1,      0,      0,,             0,,             0,      ;
break_out,      1,      0,      0,,             0,,             1,      ;
break_off,      0,      1,      1,,             1,,             1,      ;
call_in,        1,      0,      1,,             1,,             1,      ;
kick_off,       1,      0,      0,,             0,,             0,      ;
kick_in,        0,      1,      0,,             0,,             0,      ;
ward_off,       1,      0,      0,,             0,,             1,      ;
wrap_up,        1,      0,      0,,             0,,             0,      ;
play_down,      1,      0,      0,,             0,,             0,      ;
roll_out,       1,      0,      0,,             0,,             1,      ;
roll_over,      1,      0,      0,,             0,,             1,      ;
cash_in,        0,      1,      0,,             0,,             0,      ;
bog_down,       1,      0,      0,,             0,,             0,      ;
hold_up,        1,      0,      0,,             0,,             0,      ;
hold_down,      1,      0,      1,,             1,,             1,      ;
hold_off,       1,      0,      1,,             1,,             1,      ;
hold_out,       0,      1,      0,,             0,,             0,      ;
chalk_up,       1,      0,      0,,             0,,             0,      ;
shrug_off,      1,      0,      0,,             0,,             0,      ;
auction_off,    1,      0,      0,,             1,,             0,      ;
pull_back,      1,      0,      1,,             1,,             1,      ;
buy_up,         1,      0,      0,,             1,,             0,      ;
```

```
buy_out,          1,      0,      0,,             1,,             0,      ;
drag_down,        1,      0,      0,,             0,,             1,      ;
check_out,        0,      1,      0,,             0,,             0,      ;
round_up,         1,      0,      0,,             0,,             0,      ;
draw_up,          1,      0,      0,,             0,,             0,      ;
go_up,            0,      1,      1,,             1,,             1,      ;
go_down,          0,      1,      1,,             1,,             1,      ;
go_ahead,         0,      1,      0,,             0,,             0,      ;
go_away,          0,      1,      1,,             1,,             1,      ;
go_on,            0,      1,      0,,             0,,             0,      ;
go_out,           0,      1,      1,,             1,,             1,      ;
go_through,       1,      0,      0,,             0,,             0,      ;
open_up,          1,      1,      0,,             1,,             0,      ;
cover_up,         1,      0,      0,,             0,,             0,      ;
throw_away,       1,      0,      1,,             1,,             1,      ;
throw_out,        1,      0,      0,,             0,,             1,      ;
dry_up,           0,      1,      0,,             0,,             0,      ;
line_up,          0,      1,      0,,             0,,             0,      ;
set_up,           1,      0,      0,,             0,,             0,      ;
set_off,          1,      0,      0,,             0,,             0,      ;
rack_up,          1,      0,      0,,             0,,             0,      ;
fall_through,     0,      1,      0,,             0,,             0,      ;
single_out,       1,      0,      0,,             0,,             0,      ;
sort_out,         1,      0,      0,,             1,,             0,      ;
figure_out,       1,      0,      0,,             1,,             0,      ;
push_up,          1,      0,      1,,             1,,             1,      ;
wind_up,          1,      1,      0,,             0,,             0,      ;
wind_down,        0,      1,      0,,             0,,             0,      ;
scare_off,        1,      0,      1,,             1,,             1,      ;
point_out,        1,      0,      0,,             0,,             0,      ;
sit_down,         0,      1,      1,,             1,,             1,      ;
settle_down,      0,      1,      0,,             1,,             0,      ;
wipe_out,         1,      0,      0,,             0,,             0,      ;
pass_up,          1,      1,      0,,             1,,             0,      ;
pass_over,        1,      0,      0,,             0,,             0,      ;
```

```
phase_out,      1,      0,      0,,             0,,             1,      ;
touch_off,      1,      0,      0,,             0,,             0,      ;
lay_off,        1,      0,      0,,             0,,             0,      ;
lay_out,        1,      0,      0,,             0,,             0,      ;
rule_out,       1,      0,      0,,             0,,             1,      ;
bring_out,      1,      0,      1,,             1,,             1,      ;
bring_in,       1,      0,      1,,             1,,             1,      ;
bring_down,     1,      0,      0,,             0,,             1,      ;
show_off,       1,      1,      0,,             1,,             0,      ;
drop_out,       0,      1,      0,,             0,,             1,      ;
snap_up,        1,      0,      0,,             0,,             0,      ;
take_away,      1,      0,      1,,             1,,             1,      ;
turn_away,      1,      0,      0,,             0,,             1,      ;
send_back,      1,      0,      1,,             1,,             1,      ;
give_away,      1,      0,      1,,             1,,             1,      ;
carve_out,      1,      0,      0,,             0,,             0,      ;
buy_back,       1,      0,      1,,             1,,             1,      ;
set_aside,      1,      0,      0,,             0,,             0,      ;
push_through,   1,      0,      1,,             1,,             1,      ;
lock_in,        1,      0,      0,,             0,,             0,      ;
knock_out,      1,      0,      0,,             0,,             0,      ;
come_down,      0,      1,      0,,             0,,             1,      ;
come_back,      0,      1,      1,,             1,,             1,      ;
pay_down,       1,      0,      1,,             1,,             1,      ;
```

Bibliography

Baldwin, T. and Villavicencio, A. (2002). Extracting the unextractable: A case study on verb-particles. In *Proc. of the 6th Conference on Natural Language Learning (CoNLL-2002)*.

Bame, K. (1999). Aspectual and resultative verb-particle constructions with up. Handout for talk presented at the Ohio State University Linguistics Graduate Student Colloquium.

Bolinger, D. (1971). *The Phrasal Verb in English*. Harvard University Press, Harvard, USA.

Briscoe, T. and Carroll, J. (2002). Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504.

Briscoe, T. and Copestake, A. (1999). Lexical rules in constraint-based grammars. *Computational Linguistics*, 25(4):487–526.

Budanitsky, A. and Hirst, G. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on Wordnet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, USA.

Calzolari, N., Fillmore, C., Grishman, R., Ide, N., Lenci, A., Macleod, C., and Zampolli, A. (2002). Towards best practice for multiword expressions in computational lexicons. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*.

Clark, S. (2001). *Class-Based Statistical Models for Lexical Knowledge Acquisition*. PhD thesis, University of Sussex.

Copestake, A. (2001). The semi-generative lexicon: limits on lexical productivity. In *Proceedings of the 1st International Workshop on Generative Approaches to the Lexicon. Geneva*.

Copestake, A. and Lascarides, A. (1997). Integrating symbolic and statistical representations: The lexicon pragmatics interface. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the EACL (ACL-EACL'97)*, pages 136–43, Madrid, Spain.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6).

Fairclough, N. (1965). Some english phrasal types: Studies in the collocation of lexical items with prepositions and adverbs in a corpus of spoken and written present-day english. Master's thesis, University College London.

Fraser, B. (1976). *The Verb-Particle Combination in English*. The Hague: Mouton.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Extractions*. Kluwer Academic.

Gries, S. (2000). *Towards multifactorial analyses of syntactic variation: The case of particle placement*. PhD thesis, University of Hamburg.

Grover, C., Carroll, J., and Briscoe, E. (1993). The alvey natural language tools grammar. Technical Report 284, Computer Laboratory, University of Cambridge.

Huddleston, R. and Pullum, G. (2002). *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, UK.

Jiang, J. and Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of of ROCLING X*.

Kennedy, A. (1920). *The Modern English Verb-Adverb Combination*. Stanford, CA: Stanford University Press.

Langacker, R. (1982). Space grammar, analysability and the english passive. *Language 58, 22-80*.

Langacker, R. (1991). *Foundations of Cognitive Grammar Vol.1: Theoretical Prerequisites*. Stanford CA, Stanford University Press.

Levin, B. (1993). *English Verb Classes and Alterations*. University of Chicago Press, Chicago, USA.

Li, H. and Abe, N. (1996). Clustering words with the mdl principle. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*.

Li, H. and Abe, N. (1998). Generalising case frames using a thesaurus and the mdl principle. *Computational Linguistics*, 24(2):217–44.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics (COLING/ACL-98)*.

Lin, D. (1998b). Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation (LREC 1998)*.

Lin, D. (1999). Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 317–24, College Park, USA.

Lindner, S. (1983). *A Lexico-semantic Analysis of Verb-particle Constructions with Up and Out*. Indiana University Linguistics Club.

Lipka, L. (1972). *Semantic Structure and Word-Formation: Verb-Particle Constructions in Contemporary English*. Wilhelm Fink Verlag.

Live, A. (1965). The discontinuous verb in english. *WORD*, 21:428–451.

Manning, C. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, USA.

McCarthy, D. (2001). *Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences*. PhD thesis, University of Sussex.

McCarthy, D., Carroll, J., and Preiss, J. (2001). Disambiguating noun and verb senses using automatically acquired selectional preferences. In *Proceedings of the SENSEVAL-2 Workshop at ACL/EACL'01*.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–44.

Miller, G. and Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Moon, R., editor (2000). *Collins Cobuild Dictionary of Phrasal Verbs*. Harper Collins.

M.P. Marcus, B. S. and Marcinkiewicz, M. (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–30.

Pearce, D. (2002). A comparative evaluation of collocation extraction techniques. In *Proceedings of Third International Conference on Language Resources and Evaluation*.

Pedersen, T. (2002). Distance 0.1.

Pulman, S. G. (1993). The recognition and interpretation of idioms. In Cacciari, C. and Tabossi, P., editors, *Idioms: Processing, Structure and Interpretation*, chapter 11. Lawrence Erlbaum Associates, Hillsdale, NJ.

Sag, I., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*.

Schutze, H. (1992). Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*.

Villavicencio, A. and Copestake, A. (2002). Phrasal verbs and the LinGO-ERG. *LinGO Working Paper No. 2002-01*.

Witten, I. and Frank, E. (2000). *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers.