Deep Lexical Acquisition of Verb Particle Constructions

Timothy Baldwin

CSLI, Stanford University
210 Panama Street, Stanford, CA 94305-4115 USA

and

Department of Computer Science and Software Engineering
University of Melbourne, Victoria 3010 Australia

Abstract

This paper proposes a range of techniques for extracting English verb–particle constructions from raw text corpora, complete with valence information. We propose four basic methods, based on the output of a POS tagger, chunker, chunk grammar and dependency parser, respectively. We then present a combined classifier which we show to consolidate the strengths of the component methods.

Key words: lexical acquisition, subcategorisation learning, verb particle

1 Introduction

There is growing awareness of the pervasiveness and idiosyncrasy of **multiword expressions** (MWEs), and the need for a robust, structured handling thereof (Sag et al., 2002; Calzolari et al., 2002). Examples of MWEs are lexically fixed expressions (e.g. *ad hoc*), idioms (e.g. *see double*), light verb constructions (e.g. *make a mistake*) and institutionalised phrases (e.g. *kindle excitement*).

MWEs pose a challenge to NLP due to their syntactic and semantic idiosyncrasies, which are often unpredictable from their component parts. Large-scale manual annotation of MWEs is infeasible due to their sheer volume (estimated to be at least equivalent to the number of simplex words (Jackendoff, 1997)),

Email address: tim@csse.unimelb.edu.au (Timothy Baldwin).

productivity and domain-specificity. This points to the need for automatic MWE extraction techniques, which is the subject of this research.

The particular MWE type we target for extraction is the English **verb-particle construction**, or VPC. VPCs consist of a head verb and one or more obligatory **particles**, in the form of intransitive prepositions (e.g. *hand in*), adjectives (e.g. *cut short*) or verbs (e.g. *let go*) (Villavicencio and Copestake, 2002; Huddleston and Pullum, 2002); for the purposes of this paper, we will focus exclusively on prepositional particles—by far the most common and productive of the three types. We define VPCs to optionally select for an NP complement, i.e. to occur both transitively (e.g. *hand in the paper*) and intransitively (e.g. *battle on*). In terms of encoding VPCs in the LinGO-ERG, transitive VPCs correspond to the `v_particle_np_le` lexical type, and intransitive VPCs to the `v_particle_le` lexical type (see Villavicencio and Copestake (2002) for further details).

The purpose of this research is to extract fully-specified VPC lexical entries from raw text corpora. Having said this, our aim is certainly not to generate an exhaustive all-purpose list of VPCs. Such an endeavour is hampered by the high productivity and domain variability of VPCs, as illustrated poignantly by Villavicencio who found that the degree of agreement in VPC content between (putatively general-purpose) static dictionaries is remarkably low (Villavicencio, 2003). It is possible, however, to pre-compile a lexicon of VPCs from a fixed corpus to allow us to pre-tune our grammar to it prior to deployment. It is this task of corpus-attuned VPC extraction that we target.

A secondary application of this research is in the detection of VPC tokens to provide the feedstock data for research on the syntax and semantics of VPCs. Indeed, this work was partly driven by independent research on VPC compositionality (Bannard et al., 2003) using distributional models of word similarity, and the realisation that transitive and intransitive VPCs often have markedly different semantics (cf. *Kim and Sandy made out* vs. *Kim made out Sandy's figure in the mist*). If we can determine the valence of each VPC token, it becomes possible to discriminate senses which fall along valence lines, hence benefitting our distributional models.

This paper is based squarely on the work of Baldwin and Villavicencio (2002), who extract VPCs from corpus data using the same basic techniques as are described in this paper. A significant divergence over this earlier research is that we combine simple VPC extraction with subcategorisation learning, in predicting the valence (intransitive or transitive) of each VPC as part of the extraction process. This research is carried out under the umbrella of **deep lexical acquisition**, that is the learning of lexical items in a form which can be fed directly into a deep grammar or other richly-annotated lexical resource. The particular deep grammar we target is the LinGO English Resource Grammar (LinGO-ERG), a medium-coverage HPSG under development at CSLI (Flickinger, 2002).

One aspect of VPCs that makes them difficult to extract is that the verb and particle can be non-contiguous, e.g. *hand the paper in* and *battle right on*. This sets VPCs apart from conventional collocations and terminology (see, e.g., McKeown and Radev (2000)) in that they cannot be captured effectively using n-grams, due to the variability in the number and type of words potentially interceding between the verb and particle.

There is considerable scope for research on learning the semantics of VPCs—or alternatively individually learning the semantics of the verb and particle in a given VPC—which would equally come under the umbrella of deep lexical acquisition. As the first step in this direction, a number of computational models of VPC compositionality have recently been proposed (Bannard et al., 2003; McCarthy et al., 2003; Schulte im Walde, 2004). For our purposes, however, there is no immediate need for VPC semantics within the LinGO-ERG (our touchstone lexical resource), and we thus focus exclusively on syntactic deep lexical acquisition in this paper.

In the remainder of this paper, we outline linguistic features of VPCs relevant to the extraction process (Section 2) and the resources used in this research (Section 3). We then present and evaluate a number of simple methods for extracting VPCs based on, respectively, POS tagging (Section 4), the output of a full text chunk parser (Section 5), a chunk grammar (Section 6) and a parser (Section 7). Finally, we detail enhancements to the basic methods (Section 8), describe secondary evaluation over the Brown corpus (Section 9), and give a brief description of related research (Section 10).

2 Linguistic Features of VPCs

Given an arbitrary verb–preposition pair, where the preposition is governed by the verb, a number of analyses are possible. If the preposition is intransitive, a VPC results. If the preposition is transitive, it must select for an NP, producing either a **prepositional verb** (e.g. *refer to*) or a **free verb–preposition combination** (e.g. *put it on the table*, *climb up the ladder*).

A number of diagnostics can be used to distinguish VPCs from both prepositional verbs and free verb–preposition combinations (Huddleston and Pullum, 2002):

(1) transitive VPCs undergo the particle alternation
(2) with transitive VPCs, pronominal objects must be expressed in the “split”

configuration
(3) manner adverbs cannot occur between the verb and particle

The first two diagnostics are restricted to transitive VPCs, while the third applies to both intransitive and transitive VPCs.

The first diagnostic is the canonical test for particlehood, and states that transitive VPCs take two word orders: the **joined** configuration whereby the verb and particle are adjacent and the NP complement follows the particle (e.g. *hand in the paper*), and the **split** configuration whereby the NP complement occurs between the verb and particle (e.g. *hand the paper in*). Note that prepositional verbs and free verb–preposition combinations can occur only in the joined configuration (e.g. *refer to the book* vs. **refer the book to*). There are also small numbers of VPCs which do not readily undergo the particle alternation, including *carry out (a threat)* (cf. *?carry (a threat) out*).

The second diagnostic stipulates that pronominal NPs can occur only in the split configuration (*hand it in* vs. **hand in it*). Note also that heavy NPs tend to occur in the joined configuration, and that various other factors interact to determine word order (see, e.g., Dehé (2002) and Wasow (2002)).

The third diagnostic states that manner adverbs cannot intercede between the verb and particle (e.g. **hand quickly the paper in*). Note that there is a small set of non-manner adverbs which can pre-modify particles (i.e. *back, right, straight, way* and *well*, such as in *jump well up*).

We recognise that VPCs can occur with lexical types other than simple intransitive and transitive, e.g. *scream out* [$_{\text{CP}}$ *that Elvis lives on*] and *burst out* [$_{\text{VP}}$ *laughing*]. In this paper, however, we focus on intransitive and transitive VPCs due to their high type and token frequency compared to other lexical types, and also because we believe that they are the lexical types with the highest level of variability and analytical ambiguity.

3 Resources

In this section, we detail the resources used in the extraction experiments, and describe a basic corpus study of the token frequency of VPCs.

3.1 Corpora and gold-standard VPC annotation

Our extraction experiments are targeted primarily at the written portion of the British National Corpus (BNC: Burnard (2000)), as we consider it a potentially rich source of VPCs due to its size (90m words) and variability of language. In order to generate gold-standard data on which to evaluate our extraction method, we first combined the VPC content of COMLEX 3.0 (Grishman et al., 1998), the Alvey Natural Language Tools grammar (Grover et al., 1993) and the LinGO-ERG, for a total of 3,204 VPCs. From this, we randomly selected 1,000 VPCs, and had an expert annotator browse through the BNC data for both intransitive and transitive token instances of each VPC type. The VPC identification tool was provided in the form of a concordancer which first presented instances of VPCs identified by the method of Baldwin and Villavicencio (2002), and then any lines in which the verb and particle co-occurred within a window of 10 words. The annotator was instructed to look through all the concordance lines until at least one instance of each valence (intransitive and transitive) of a given VPC was detected, or alternatively the concordance data ran out.

We contrast our proposed corpus attestation-based evaluation method to the research of Baldwin and Villavicencio (2002), e.g., who evaluate precision based on a fixed lexical resource. As VPCs are highly productive and involve commonly-occurring words (verbs and prepositions), it is relatively easy to find words in a given combination, without any guarantee that they occur in a VPC. It is thus vital that we safeguard against false positive hits with a static dictionary by validating the token evidence for each VPC candidate, which is what we claim our evaluation method does.

The conditioning of our sample VPC data on a static dictionary is simply in the interests of achieving a representative evaluation window. That is, it is a convenient way of fine-tuning the extraction method in a way which we can expect to scale to the open set of verb–preposition combinations. In terms of applying the proposed method in a real-world VPC extraction task, what verb–particle combinations are considered as VPC candidates is conditioned purely on the composition of the corpus, and the random set of 1,000 VPCs is only relevant in terms of providing training data.

In addition to the BNC data, we used the Brown and Wall Street Journal (WSJ) corpora—as contained in the Penn Treebank (Marcus et al., 1993)—as training data, and thus had the annotator replicate the VPC annotation task over these two corpora. In this, we deliberately chose not to use the gold-standard annotations found in the treebank as we found them to be inconsistent, particularly for intransitive VPCs. In the Brown corpus, e.g., there are two occurrences of the intransitive *mess around*, with *around* tagged as a particle in one and an adverb in the other. There are also 8 occurrences of the intransitive *walk around*, in which *around* is tagged as a transitive(!) preposition 4 times and an adverb 4 times.

Corpus	***Attested LEs***	***Mean frequency***	***Median frequency***
Brown	21.4%	2.3	1
WSJ	21.2%	3.4	1
BNC	69.9%	89.6	7

Table 1
Statistical analysis of VPC corpus occurrence

3.2 VPC corpus occurrence

In order to get a feel for the relative frequency of VPCs in corpus data, we took our random sample of 1,000 VPCs and did a semi-automatic[1] corpus search for each in the Brown, WSJ and BNC data, as detailed in Table 1.

The coverage of VPC types for the BNC is considerably greater than the Brown and WSJ corpora, as is the mean token frequency. The most telling statistic, however, is the median, which tells us that at least half of the Brown and WSJ VPCs occur only once, and in the case of the BNC, 7 or less times. This illustrates the severity of data sparseness in the VPC extraction task.

4 Tagger-based Extraction

One obvious method for extracting VPCs is to run a simple regular expression over the output of a part-of-speech (POS) tagger, based on the observation that the Penn Treebank POS and CLAWS2 tagsets, e.g., contain a dedicated particle tag (`RP`). Given that all particles are governed by a verb, extraction consists of simply locating each particle and searching back (to the left of the particle, as particles cannot be passivised or otherwise extraposed) for the head verb of the VPC. The process is illustrated by the input *Cady stuck his jaw out*, for which the lemmatised Penn-style tagger output would be:

```
Cady_NNP stick+ed_VBD his_PRP$ jaw_NN out_RP
```

allowing us to read off the particle *out* and governing verb *stick* directly.

Here and for the subsequent methods, we assume that the maximum word length for NP complements in the split configuration for transitive VPCs is 5,[2] i.e. that an NP "heavier" than this would occur more naturally in the

[1] Based on weighted voting across token occurrences identified by our extraction methods, tuned to align with the results of a manual token count of VPCs in the WSJ corpus.
[2] Note, this is the same as the maximum span length of 5 used by Smadja (1993), and above the maximum attested NP length of 3 observed in the corpus study of

joined configuration (cf. *?Sandy handed the paper which Kim gave her in* vs. *Sandy handed in the paper which Kim gave her*). We thus discount all particles which are more than 5 words from their governing verb, and additionally stipulate that the only POS types that can occur between a verb and its particle are nouns, pronouns, adjectives and determiners. To further constrain the extraction process, we extracted a set of 35 canonical particles from the 3,204 VPCs found in the combined dictionaries (see Appendix A), and used this to filter out extraneous particles in the POS data.

Having detected the VPC token instances, we perform a crude valence classification by: (a) in the case that the VPC is split, checking for an NP analysis of the words intervening between the particle and verb; and (b) in the case that the VPC is joint, checking for an NP analysis of the words immediately following the particle. Our NP analysis takes the form of a crude regular expression over the tag sequence in question. This simplistic method clearly favours a transitive VPC analysis.

The output of the extraction process takes the form of two values for each VPC: the raw frequency of intransitive and transitive VPC tokens, respectively, detected for that VPC.[3] We build a supervised classifier from the resultant 2-tuples using the TiMBL memory-based learner (Daelemans et al., 2003), taking the training data from the Brown and WSJ corpora and the test data from the BNC. The positive training exemplars for a given classification task are those VPC types out of our 1,000 VPC sample which are attested with the given valence in the training corpora (Brown and WSJ), and the negative training exemplars are conversely those in the sample which are not attested with the given valence. Evaluation is performed according to stratified 10-fold cross validation, with the slight twist that on each iteration, the test data is made up of the feature vectors for the 100 held-out VPCs based on the BNC data, and the training data is made up of the feature vectors for each of the 900 remaining VPCs in each of the Brown and WSJ corpora (i.e. the total number of training exemplars is 1,800 on each iteration).

In line with our assumption of raw text to extract over, we automatically tagged the three corpora, for which purpose we individually tested a total of three taggers: the HMM-based CLAWS2 tagger distributed with the RASP toolkit (Briscoe and Carroll, 2002), a Penn tagset-based tagger custom built using fnTBL 1.0 (Ngai and Florian, 2001), and mxpost, a maximum entropy tagger based on the Penn tagset (Ratnaparkhi, 1996).[4] Our reasons for choos-

split VPCs by Baldwin and Villavicencio (2002).

[3] Normalisation of frequency by way of negative log-likelihood was tested, but found to degrade classification performance.

[4] Note that the CLAWS2 tagger was trained over the BNC, and the fnTBL tagger and mxpost over the Penn Treebank. Thus, each tagger is used to tag at least one dataset it has been trained over and one it has not.

ing this particular assortment of taggers related to algorithmic (HMM vs. transformation-based learning vs. maximum entropy) and tagset differentiation (CLAWS2 vs. Penn), and also availability. We further lemmatised the data in each case using morph (Minnen et al., 2001).

The results for tagger-based extraction are presented in Table 2, evaluated according to the standard measures of precision, recall and F-score. Here and for the remainder of the paper, we break down evaluation into three binary classification tasks: valence-underspecified VPC extraction (**No valence**), intransitive VPC extraction (**Intransitive**) and transitive VPC extraction (**Transitive**). Note that we use the exact same feature set for each extraction task.

The results for valence-underspecified VPC extraction are surprisingly good, with very high precision but considerably lower recall. Those for intransitive and transitive VPCs, on the other hand, are somewhat disappointing. The principal reason for transitive VPC extraction performing so much better than intransitive VPC extraction is that our method explicitly favours a transitive analysis. The reason that the recall is relatively low in each case is that it is notoriously difficult for POS taggers to differentiate between particles, transitive prepositions and adverbs (Toutanova and Manning, 2000), and the default tag in instances of ambiguity tends to be a transitive preposition or adverb rather than a particle.

5 Chunker-based Extraction

To overcome the shortcomings of the taggers in identifying particles and the difficulties in determining valence from the tagger output, we next look to full text chunk parsing. Full text chunk parsing involves partitioning up a text into syntactically-cohesive, head-final segments ("chunks"), without attempting to resolve inter-chunk dependencies. In the chunk inventory devised for the CoNLL-2000 test chunking shared task (Tjong Kim Sang and Buchholz, 2000), a dedicated particle chunk type once again exists (`RP`). It is therefore possible to adopt an analogous approach to that for tagger-based extraction, in identifying particle chunks and looking back for the associated verb.

The chunk parser used in this research was custom built using fnTBL, and run over the lemmatised output of the fnTBL tagger. The parser was trained over WSJ and Brown data generated with the aid of the conversion script from the CoNLL-2000 shared task.

We extracted VPCs from the chunker output by identifying each particle chunk, and searching back for the governing verb. As for tagger-based extraction, we allow a maximum of 5 words to intercede between a particle and its

Method		*No valence*			*Intransitive*			*Transitive*		
		P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Tagger	CLAWS2	.971	.740	.840	.605	.532	.566	.834	.714	.770
	fnTBL	.979	.830	.898	.575	.449	.505	.894	.783	.835
	mxpost	.973	.881	.925	.572	.449	.504	.887	**.829**	**.857**
Chunker		**.991**	.740	.847	.599	.493	.541	**.935**	.733	.822
Chunk grammar		.984	**.897**	**.939**	**.671**	**.852**	**.751**	.888	.828	**.857**
Parser		.975	.721	.829	.636	.664	.650	.860	.712	.779
Combined		.976	.963	.969	.654	.877	.749	.892	.901	.897

Table 2
VPC extraction results over the BNC (***P*** = precision, ***R*** = recall, ***F*** = F-score; the highest score for the individual methods in each column is given in **bold**)

governing verb, and we apply the additional stipulation that the only chunks that can occur between the verb and the particle are: (a) noun chunks, (b) preposition chunks adjoining two noun chunks, and (c) adverb chunks found in our closed set of particle pre-modifiers (see Section 2). Additionally, we once again used the gold standard set of 35 particles to filter out extraneous particle chunks.

To illustrate this process, the lemmatised chunker output for *Cady stuck his jaw out* would be:

$[_{NP}$ `Cady_NNP]` $[_{VP}$ `stick+ed_VBD]` $[_{NP}$ `his_PRP jaw_NN]` $[_{RP}$ `out_RP]`

from which we are able to unambiguously identify a transitive occurrence of *stick out*.

All VPC token candidates that satisfy the above requirements are subclassified as: (1) conforming to the linguistic tests for intransitive/transitive VPCs (including looking to the right of the particle chunk to look for clause boundaries), or (2) being ambiguous, in which case we apply a set of valence determination heuristics similar to the taggers. From this we derive 7 feature values per VPC: (1) the raw token frequency of each of these subclasses ($\times 4$); (2) the ratio of intransitive and transitive VPC tokens to the total token count ($\times 2$); and (3) the mutual information between VPC tokens and the verb and particle. Once again, we build a supervised classifier with TiMBL, and evaluate using 10-fold stratified cross validation.

The precision for chunker-based extraction (see Table 2) over the valence-underspecified and transitive subtasks is significantly better than for tagger-based extraction, although the recall figures are marginally down in all but the intransitive VPC classification task. Chunking and our stipulations on what can occur between a verb and particle thus appear to over-constrain the search space for VPCs.

6 Chunk Grammar-based Extraction

The principle weakness of chunker-based extraction was recall, leading us to implement a rule-based chunk sequencer which searches for particles in prepositional and adverbial chunks as well as particle chunks. In essence, we take each verb chunk in turn, and search to the right for a single-word particle, prepositional or adverbial chunk which is contained in the gold standard set of 35 particles. For each such chunk pair, we then analyse: (a) the chunks that occur between them to ensure that, maximally, an NP and particle premodifier adverb chunk are found; (b) the chunks that occur immediately after the particle/preposition/adverb chunk to check for a clause boundary or NP; and (c) the clause context of the verb chunk for possible extraposition of an NP verbal complement, through passivisation or relativisation.

The objective of this analysis is to both determine the valence of the VPC candidate (intransitive or transitive) and identify evidence either supporting or rejecting a VPC analysis relative to the known linguistic properties of VPCs, as described in Section 2. For example, if a pronominal noun chunk were found to occur immediately after the (possible) particle chunk (e.g. **see off him*), a VPC analysis would not be possible. Alternatively, if a clause boundary (e.g. a full stop) were found to occur immediately after the "particle" chunk in a non-passivised clause and nothing interceded between the verb and particle chunk, then this would be evidence for an intransitive VPC analysis.

The chunk sequencer is not able to furnish positive or negative evidence for a VPC analysis in all cases. Indeed, in a high proportion of instances, a noun chunk was found to follow the "particle" chunk, leading to ambiguity between analysis as a VPC, prepositional verb or free verb–preposition combination (see Section 2), or in the case that an NP occurs between the verb and particle, the "particle" being the head of a PP post-modifying an NP. As a case in point, the VP *hand the paper in here* could take any of the following structures: (1) *hand* [$_{\mathsf{NP}}$ *the paper*] [$_{\mathsf{RP}}$ *in*] [$_{\mathsf{NP}}$ *here*] (transitive VPC *hand in* with adjunct NP *here*), (2) *hand* [$_{\mathsf{NP}}$ *the paper*] [$_{\mathsf{PP}}$ *in here*] (transitive prepositional verb *hand in* or simple transitive verb with PP adjunct), and (3) *hand* [$_{\mathsf{NP}}$ *the paper in here*] (simple transitive verb). In such cases, we resolve the attachment ambiguity using an unsupervised disambiguation method based on the log-likelihood ratio ("LLR", Dunning (1993)). That is, we use the chunker output to enumerate all the verb–preposition, preposition–noun and verb–noun bigrams in each corpus, based on chunk heads rather than strict word bigrams. We then use frequency data to pre-calculate the LLR for each such type. In the case that the verb and "particle" are joined (i.e. no NP occurs between them), we simply compare the LLR of the verb–noun and particle–noun pairs, and assume a VPC analysis in the case that the former is strictly larger than the latter. In the case that the verb and "particle" are split (i.e. we have the chunk sequence $\texttt{VC NC}_1\ \texttt{PC NC}_2$),[5] we calculate three scores: (1) $\text{LLR}(\texttt{VC}, \texttt{PC}) \times \text{LLR}(\texttt{VC}, \texttt{NC}_2)$, (2) $\text{LLR}(\texttt{VC}, \texttt{PC}) \times \text{LLR}(\texttt{PC}, \texttt{NC}_2)$, and (3) $\text{LLR}(\texttt{NC}_1, \texttt{PC}) \times \text{LLR}(\texttt{PC}, \texttt{NC}_2)$; these scores model the plausibility of each of the three structures posited for *hand the paper in here* above. Only in the case that the first of these is strictly greater than the other two, do we favour a (transitive) VPC analysis.

As an example of this process, consider the input *Kim flops around all day* and the chunker output:

$[_{NP}$ `Kim_NNP]` $[_{VP}$ `flop+s_VBZ]` $[_{PP}$ `around_IN]` $[_{NP}$ `all_DT day_NN]`

in which *around* has been analysed as a transitive preposition chunk. Our chunk grammar would identify *around* as a potential particle and check the clause against our linguistic tests. No positive or negative linguistic test applies in this case, forcing us to fall back on LLR-based disambiguation, which leads to the reclassification of *around* as a particle.

Based on the positive and negative grammatical evidence from above, for both intransitive and transitive VPC analyses, we generate four frequency-based features. The advent of data derived through attachment resolution, again for both intransitive and transitive VPC analyses, provides another two features.

Analogously to the preceding methods, we build our classifier using TiMBL and evaluated based on 10-fold stratified cross-validation, the results for which are presented in Table 2. The chunk grammar produces the highest overall F-score for all extraction tasks, and is markedly better at the intransitive VPC extraction task than the taggers and chunker. As with the previous methods, there is a noticeable decrement in performance through the valence-underspecified, transitive and intransitive extraction tasks.

7 Parser-based Extraction

Following on the natural path of evolution in terms of the linguistic sophistication of the pre-processor used in extraction, we finally turn to extraction based on a full parser. The particular parser we target is RASP (Briscoe and Carroll, 2002), due to its robustness over different text types and ability to analyse

[5] Here, `VC` = verb chunk, `NC` = noun chunk and `PC` = (intransitive or transitive) preposition chunk.

phrase structure through a tag sequence grammar. Recall that the CLAWS2 tagger used above (Section 4) was that employed in the RASP system.

The RASP-based extraction process can be illustrated with our example input *Cady stuck his jaw out*, for which the output is:

```
(|ncsubj| |stick+ed:2_VVD| |Cady:1_NP1| _)
(|dobj| |stick+ed:2_VVD| |jaw:4_NN1| _)
(|detmod| |poss| |jaw:4_NN1| |his:3_APP$|)
(|mod| _ |stick+ed:2_VVD| |out:5_RP|)
```

The first field in each tuple is the grammatical relation type (e.g. `ncsubj` = subject, `dobj` = direct object), which is followed by the head and modifier, respectively, each of which is given as the word lemma, position index in the input (e.g. word 5 in the case of *out*) and CLAWS2 POS tag. VPCs are marked by the `mod` relation, and the fact that the head is a verb and the modifier a particle (`RP`). We can combine this with valence information for the verb to trivially read off the fact that, in the case above, *stick out* is a transitive VPC.

In the case that no spanning parse is found for the tagged input, RASP flags the output and returns partial parse information. In identifying token instances of a given VPC, therefore, we count the number of times RASP identifies that VPC as occurring with each of the two valences, in the full and partial parse data. We thus generate a total of 4 features, which we once again feed into TiMBL and evaluate via 10-fold stratified cross-validation (see Table 2).

The results for the valence-underspecified VPC extraction task are below those of all other methods, but RASP comes into its own in determining valence, particularly for the intransitive task. These figures are misleading, however, as RASP only recognises 9 of the 35 particles targeted in this research (see Section A for details), which account for only 744 of the 1,000 VPCs in our random sample; these 9 particles include the 7 most common particles in the Brown and WSJ corpora, and account for an average of 95.3% of all particle token instances (according to the Penn Treebank gold-standard annotation). If we limit evaluation to only these 744 VPCs, we find that the relative RASP and CLAWS2 tagger performance is considerably improved, as detailed in Table 3. Precision is largely unchanged, but, unsurprisingly, recall receives a significant boost, and RASP surpasses all other methods in F-score over the intransitive task. Likewise, the recall of the other taggers and the chunker improves appreciably while precision stays roughly constant. The performance of the chunk grammar, meanwhile, is largely unchanged. It is to be expected that taggers and chunker should also perform better over this subset of particles, given their high frequency within the Penn treebank data over which they were trained. More encouraging is the finding that the chunk grammar method performs approximately as well over high and low frequency particles.

Method		***No valence***			***Intransitive***			***Transitive***		
		P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Tagger	CLAWS2	.971	**.978**	**.975**	.602	.726	.658	.834	**.876**	.854
	fnTBL	.977	.934	.955	.545	.489	.516	.900	.861	.880
	mxpost	.973	.971	.972	.541	.491	.515	.891	.874	**.883**
Chunker		**.990**	.846	.912	.592	.580	.586	**.941**	.790	.859
Chunk grammar		.984	.903	.942	**.643**	.857	.734	.861	.858	.859
Parser		.975	.952	.964	.638	**.914**	**.751**	.859	.874	.866
Combined		.976	.990	.983	.628	.928	.749	.898	.922	.910

Table 3
VPC extraction results over the BNC for only the RASP particles

Method		***No valence***			***Intransitive***			***Transitive***		
		P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Tagger	CLAWS2	5.7	2.7	3.0	3.7	4.3	4.3	6.0	4.0	5.0
	fnTBL	2.7	2.7	2.7	4.3	3.7	4.0	2.7	**2.0**	**2.0**
	mxpost	3.3	**1.7**	**1.7**	5.7	3.0	3.3	3.0	2.7	2.3
Chunker		**1.0**	5.0	4.7	4.0	5.3	5.3	**2.0**	2.7	2.7
Chunk grammar		4.3	4.0	4.0	**1.0**	**2.3**	**1.7**	2.3	4.3	4.0
Parser		4.0	5.0	5.0	2.3	**2.3**	2.3	5.0	5.3	5.0

Table 4
Mean rank of the individual methods for VPC extraction over all three corpora

8 Improving on the Basic Methods

To get a more general picture of the strengths and weaknesses of the individual methods, we calculated the mean ranking of each method in terms of precision, recall and F-score for the three extraction tasks; the rankings are averaged across three experiments, extracting VPCs from the BNC (as above), the Brown corpus (see below) and the WSJ corpus, using the remaining two corpora as training data in each case. From this, we are able to validate our observations over the BNC, in that the taggers and chunker are remarkably effective at the valence-underspecified lexical acquisition task, the chunk grammar and RASP are superior to the other methods over the intransitive VPC extraction task, and there is considerable variance and no one single standout system over the transitive VPC extraction task.

In order to capitalise on the respective strengths of the different methods, in this section, we investigate the possibility of combining the methods into a single consolidated classifier. System combination is achieved by concatenating the feature vectors generated by the component methods.[6]

The results of this simple combination process over the BNC are presented in Table 2 (as **Combined**), based on the best-performing configuration of each basic extraction method (i.e. only mxpost for tagger-based extraction, and each of chunker-, chunk grammar- and parser-based extraction). Encouragingly, recall improved significantly[7] over the best of the individual methods in each case, as did the F-score in all cases other than for intransitive extraction, where we were unable to improve upon the chunk grammar. The final F-score for the valence-underspecified task was 0.969, while that for the intransitive and transitive VPC extraction tasks were 0.749 and 0.897, respectively.

The principal reason for intransitive VPC extraction being more difficult than transitive VPC extraction is the difficulty in distinguishing between intransitive VPCs (e.g. *wash up, break down*) and intransitive simplex verbs with directional adverbs (e.g. *drop down, jump up*). There is no easy way of distinguishing the two using shallow syntactic features, and the only real way to tease them apart is in terms of whether the verb–preposition combination has compositional semantics or not. Given the relative immaturity of computational models of semantic compositionality, this goes beyond the scope of current technology. Looking through the output of the combined classifier, we tend to include more verb–adverb combinations into the lexicon than we would ideally like to (as is borne out by the diminished precision). In terms of deep grammar applications, this has little impact on performance, and the only real effect is to clutter the lexicon. Perhaps the best means of removing these from the lexicon is a method of the type suggested by Villavicencio and Copestake (2002), whereby we predict compositional directional adverb–verb combinations via preposition semantics and verb classes.

Baldwin and Villavicencio (2002) had considerable success in adding in extra features which reflected VPC canonicity, including whether a given VPC occurred in a deverbal or adjectival form in the corpus (e.g. *turnaround*, *dried-up*). We similarly tried enhancing the combined classifier with a range of such features, but found that they invariably diminished performance. The reason for this is that the classifier was led to overgeneralise and ignore token evidence, such that common but unattested VPCs found their way into the data. This is a nice illustration of how our evaluation strategy keeps us faithful to

[6] We also tried a cascaded classifier architecture, with the component classifiers feeding into a combined classifier, but found that simple feature vector concatenation produces consistently better results.

[7] Based on the paired t test ($p < 0.05$).

the target corpus.

In additional experimentation, we tried basing our training exemplar gold-standard annotations on dictionary data (using the combined COMLEX, Alvey Tools and LinGO-ERG data) rather than corpus attestations, but found that performance dropped appreciably in almost all cases. We additionally tested the complementarity of corpus-based and dictionary data, by adding in dictionary data for only those VPCs not in the 1,000 sample, but found that the impact on results was negative.

9 Extraction over the Brown Corpus

In Section 3.2 above we observed that the VPC token frequency in the three corpora used in this research was remarkably low, particularly in the Brown and WSJ corpora. To date, evaluation has been focused on extracting VPCs out of the BNC, which was shown to have a median VPC frequency of around 7. We have yet to consider whether the methods are equally suited to extracting VPCs out of corpora with lower VPC frequencies. In this section, we thus analyse VPC extraction performance over the Brown corpus (with a median VPC frequency of 1).

In targeting the Brown corpus, we are also able to investigate the correlation between the accuracy of the preprocessor systems and extraction performance as, unlike the BNC, we have access to gold-standard POS tag and chunk data to directly evaluate a large section of our preprocessors against. Our procedure for doing this is to run our extraction methods over the gold standard annotations, in an attempt to establish an upper bound on the accuracy of the methods given "perfect" preprocessor output. Specifically, we run the tagger-based extraction method over the gold-standard POS tags in the Brown and WSJ corpora, and the chunker- and chunk grammar-based extraction methods over gold-standard chunk data derived from the treebank data; for the BNC, we have no option but to use the output of our preprocessors as before (choosing mxpost to provide the "gold standard" POS tag data).[8] Note we are unable to perform this process for the parser-based extraction method as we have no way of automatically generating RASP-style gold-standard annotations. As before, we combine the training exemplars from the WSJ corpus and BNC and perform stratified 10-fold stratified cross-validation with the Brown corpus as our test data.

[8] We also carried out evaluation using only the WSJ corpus as training data, but found that results generally diminished. The general findings reported in this section were additionally validated over the WSJ corpus in an analogous extraction experiment.

Method		*No valence*			*Intransitive*			*Transitive*		
		P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Tagger	CLAWS2	.948	**.732**	**.826**	.786	.382	.514	.653	.631	.642
	fnTBL	.969	.692	.807	.776	.480	.593	.878	**.675**	**.763**
	mxpost	.970	.717	.824	.765	.509	.611	**.892**	.611	.725
	Penn	.964	.664	.786	.707	.405	.515	.899	.660	.761
Chunker	fnTBL	**.977**	.530	.687	.805	.358	.496	.868	.616	.720
	Penn	.984	.558	.712	.742	.399	.519	.871	.567	.687
Chunk grammar	fnTBL	.934	.573	.710	**.875**	.486	.625	.891	.483	.626
	Penn	.932	.773	.845	.839	.543	.660	.865	.759	.808
Parser		.959	.589	.730	.796	**.520**	**.629**	.786	.433	.559
Combined	Auto	.948	.900	.923	.776	.699	.736	.814	.798	.806
	Penn	.945	.863	.902	.759	.711	.734	.811	.739	.773

Table 5
VPC extraction results over the Brown corpus

The results for VPC extraction over the Brown corpus, using the BNC and WSJ corpus as training data, are presented in Table 5, with the results using the gold-standard Penn treebank annotations for each basic method given as **Penn**.

Overall, we find a greater spread in relative system performance than was observed in Table 2, with each system producing a highest score in at least one of the columns. Recall is generally well below that for the BNC in the case of the individual methods, but through system combination, the F-score comes right up to approximately the same levels as for the BNC. Considering that at least half of the attested VPCs only occur once in the Brown corpus, the combined method appears remarkably robust to low frequency data and adept at combining the shreds of evidence provided by the individual methods.

Comparing the results based on the gold-standard Penn data with those for the preprocessors, the biggest gain seems to be achieved for chunk grammar-based extraction, with overall marginally better results for chunker-based extraction and, surprisingly, better results for tagger-based extraction when using our taggers than the gold-standard data. The surprisingly low impact of gold-standard data on the extraction process is to some degree due to the BNC data offsetting the results: when we use only the WSJ as training data, results drop overall, but there is a more consistent—if small—increment gained from using the gold-standard data. One intriguing result is that system combination over the output of the preprocessors (**Auto** in Table 5) produces superior results to that over the gold-standard data. That is, improving the accuracy of our preprocessors, in particular the chunker, would appear to boost the performance of the component extraction methods, but system combination seems to nullify any such potential gains by compensating for noise in the output of the individual preprocessors.

10 Related research

There is a moderate amount of research related to the extraction of VPCs, or more generally phrasal verbs (i.e. VPCs and prepositional verbs), which we briefly describe here.

As alluded to variously, this paper represents a direct extension of Baldwin and Villavicencio (2002). The primary differences over this earlier paper are that we: (a) perform subcategorisation learning as part of the extraction process; (b) base evaluation fully on actual corpus occurrence, whereas Baldwin and Villavicencio used an independent list of VPCs to evaluate precision; (c) treat VPC extraction as a fully supervised task, whereas Baldwin and Villavicencio treated it as an unsupervised or weakly-supervised task; (d) evaluate the method over a large-scale corpus (the BNC), whereas Baldwin and Villavicencio focused on the WSJ; and (e) explore the possibility of parser-based extraction.

We are not the first to attempt to use RASP for the valence-underspecified VPC extraction task. McCarthy et al. (2003) used it to extract out VPC tokens from BNC data, the results of which were utilised in a MWE compositionality study. McCarthy et al. evaluated the token accuracy of their method relative to the WSJ corpus, making direct comparison with this research difficult, although their basic finding that RASP offers high-precision (0.926), medium-recall (0.642) extraction is parallelled in our valence-underspecified results (0.975 precision and 0.721 recall).

One of the earliest attempts at extracting "interrupted collocations" (i.e. non-contiguous collocations, including VPCs), was that of Smadja (1993). Smadja based his method on bigrams, but unlike conventional collocation work, described bigrams by way of the triple of $\langle word_1, word_2, posn \rangle$, where *posn* is the number of words occurring between $word_1$ and $word_2$ (up to 4). For VPCs, we can reasonably expect from 0 to 4 words to occur between the verb and the particle, leading to 5 distinct variants of the same VPC and no motivated way of selecting between them. Smadja did not attempt to evaluate his method other than anecdotally, making any comparison with our research impossible.

The work of Blaheta and Johnson (2001) is closer in its objectives to our research, in that it takes a parsed corpus and extracts out multiword verbs (i.e. VPCs and prepositional verbs) through the use of log-linear models. Once again, direct comparison with our results is difficult, as Blaheta and Johnson output a ranked list of all verb–preposition pairs, and subjectively evaluate the quality of different sections of the list. Additionally, they make no attempt to distinguish VPCs from prepositional verbs.

In a slightly different vein, Li et al. (2003) developed an expert lexicon method for identifying phrasal verb token occurrences. That is, they assume a static lexicon of phrasal verbs (both VPCs and prepositional verbs) and build templates predicting the syntax of each lexical item based largely on surface clues and shallow syntactic structure. They evaluate their method to be remarkably accurate, at an F-score of around 96%. While their method is not directly applicable to the task of detecting novel occurrences of VPCs, it has potential applications in detecting instances of a core lexicon of VPCs.

There is a considerable body of literature on subcategorisation learning for verbs, an excellent overview of which can be found in Korhonen (2002). Subcategorisation methods tend to rely on statistical methods and assume a high token count for a given verb, an assumption that we clearly cannot make with VPCs (recalling that the median frequency in the BNC was 7 and that in the Brown corpus was 1).

11 Conclusion

In conclusion, this paper has been concerned with the extraction of valence-specified English verb–particle constructions from raw text corpora. Four basic methods were proposed, based on tagger output, chunker output, a chunk grammar and parser output. We then experimented with combining the output of the four component methods together into a single classifier, culminating in an F-score of 0.749 and 0.897 for intransitive and transitive verbs, respectively, over the BNC. In secondary evaluation over the Brown corpus, we showed the combined extraction method to be highly robust over low-frequency data and effective at compensating for noise in the outputs of the preprocessors.

In future research, we are interested in extending the method to extract prepositional verbs, many of which appear as false positives in the output of the classifiers. We also wish to test our method over other VPC lexical types, so as to road test the extensibility of the proposed method.

Another direction in which this research could be extended (suggested by an anonymous reviewer) would be to feed back the results of VPC extraction to generate bootstrap POS/chunk data, and refine the particle tagging performance of our taggers and chunker.

One untapped source of evidence which we believe can boost classification accuracy is token-level agreement between the component classifiers. At present, we simply look at the relative number of token occurrences each method is able to identify, without considering whether they are the same or independent instances of the given VPC. A procedure such as that proposed by Baldwin and Bond (2003) has considerable promise in forming confidence judgements from analysis of token-level agreement.

Acknowledgements

This research was supported in part by NSF grant BCS-0094638 and also the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University. We would like to thank Colin Bannard, Francis Bond, John Carroll, Ann Copestake, Dan Flickinger, Yuji Matsumoto, Diana McCarthy, Hiromi Nakaiwa, Aline Villavicencio and the anonymous reviewers for their valuable input on this research, and Ara Kim for her patience and attention to detail in annotating the data.

A Canonical particles

The following are the 35 canonical particles used in this research, of which the 9 underlined particles are those handled by RASP:

> *about, above, abroad, across, adrift, ahead, along, apart, around, aside, astray, away, back, backward, behind, by, down, forth, forward, in, into, off, on, open, out, over, short, through, to, together, under, up, upon, with, without*

References

Baldwin, T. and F. Bond (2003). A plethora of methods for learning English countability. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, Sapporo, Japan, pp. 73–80.

Baldwin, T. and A. Villavicencio (2002). Extracting the unextractable: A case study on verb-particles. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, pp. 98–104.

Bannard, C., T. Baldwin, and A. Lascarides (2003). A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan, pp. 65–72.

Blaheta, D. and M. Johnson (2001). Unsupervised learning of multi-word verbs. In *Proceedings of the ACL/EACL 2001 Workshop on the Computational Extraction, Analysis and Exploitation of Collocations*, Toulouse, France, pp. 54–60.

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

Burnard, L. (2000). *User Reference Guide for the British National Corpus.* Technical report, Oxford University Computing Services.

Calzolari, N., C. Fillmore, R. Grishman, N. Ide, A. Lenci, C. MacLeod, and A. Zampolli (2002). Towards best practice for multiword expressions in computational lexicons. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Canary Islands, pp. 1934–40.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch (2003). *TiMBL: Tilburg Memory Based Learner, version 5.0, Reference Guide.* ILK Technical Report 03-10.

Dehé, N. (2002). *Particle Verbs in English: Syntax, Information, Structure and Intonation.* Amsterdam, Netherlands/Philadelphia USA: John Benjamins.

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

Flickinger, D. (2002). On building a more efficient grammar by exploiting types. *Journal of Natural Language Engineering,* (Special Issue on Efficient Processing with HPSG) *6*(1).

Grishman, R., C. Macleod, and A. Myers (1998). *COMLEX Syntax Reference Manual.* Proteus Project, NYU. (`http://nlp.cs.nyu.edu/comlex/refman.ps`).

Grover, C., J. Carroll, and E. Briscoe (1993). The Alvey Natural Language Tools grammar (4th release). Technical Report 284, Computer Laboratory, Cambridge University, UK.

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

Jackendoff, R. (1997). *The Architecture of the Language Faculty.* Cambridge, USA: MIT Press.

Korhonen, A. (2002). *Subcategorization Acquisition.* Ph. D. thesis, University of Cambridge.

Li, W., X. Zhang, C. Niu, Y. Jiang, and R. K. Srihari (2003). An expert lexicon approach to identifying English phrasal verbs. In *Proceedings of the 41st Annual Meeting of the ACL*, Sapporo, Japan, pp. 513–20.

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

McCarthy, D., B. Keller, and J. Carroll (2003). Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan.

McKeown, K. and D. Radev (2000). Collocations. In R. Dale, H. Moisl, and H. Somers (Eds.), *Handbook of Natural Language Processing*, Chapter 21. Marcel Dekker.

Minnen, G., J. Carroll, and D. Pearce (2001). Applied morphological processing of English. *Natural Language Engineering 7*(3), 207–23.

Ngai, G. and R. Florian (2001). Transformation-based learning in the fast lane. In *Proceedings of the 2nd Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL2001)*, Pittsburgh, USA, pp. 40–7.

Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-96)*, Philadelphia, USA.

Sag, I. A., T. Baldwin, F. Bond, A. Copestake, and D. Flickinger (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)*, Mexico City, Mexico, pp. 1–15.

Schulte im Walde, S. (2004). Identification, quantitative description, and preliminary distributional analysis of German particle verbs. In *Proceedings of the COLING Workshop on Enhancing and Using Electronic Dictionaries*, Geneva, Switzerland, pp. 85–8.

Smadja, F. (1993). Retrieving collocations from text: Xtract. *Computational Linguistics 19*(1), 143–77.

Tjong Kim Sang, E. F. and S. Buchholz (2000). Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 4th Conference on Computational Natural Language Learning (CoNLL-2000)*, Lisbon, Portugal.

Toutanova, K. and C. D. Manning (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, Hong Kong, China.

Villavicencio, A. (2003). Verb-particle constructions and lexical resources. In *Proceedings of the ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan, pp. 57–64.

Villavicencio, A. and A. Copestake (2002). Verb-particle constructions in a computational grammar of English. In *Proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar (HPSG-2002)*, Seoul, South Korea.

Wasow, T. (2002). *Postverbal Behavior*. Stanford, USA: CSLI Publications.