

# Chart Mining-based Lexical Acquisition with Precision Grammars

Yi Zhang,<sup>♠</sup> Timothy Baldwin,<sup>♡◇</sup> Valia Kordoni,<sup>♠</sup> David Martinez<sup>◇</sup> and Jeremy Nicholson<sup>♡◇</sup>

<sup>♠</sup> DFKI GmbH and Dept of Computational Linguistics, Saarland University, Germany

<sup>♡</sup> Dept of Computer Science and Software Engineering, University of Melbourne, Australia

<sup>◇</sup> NICTA Victoria Research Laboratory

yzhang@coli.uni-sb.de, tb@ldwin.net, kordoni@dfki.de,  
{davidm, jeremymn}@csse.unimelb.edu.au

## Abstract

In this paper, we present an innovative chart mining technique for improving parse coverage based on partial parse outputs from precision grammars. The general approach of mining features from partial analyses is applicable to a range of lexical acquisition tasks, and is particularly suited to domain-specific lexical tuning and lexical acquisition using low-coverage grammars. As an illustration of the functionality of our proposed technique, we develop a lexical acquisition model for English verb particle constructions which operates over unlexicalised features mined from a partial parsing chart. The proposed technique is shown to outperform a state-of-the-art parser over the target task, despite being based on relatively simplistic features.

## 1 Introduction

Parsing with precision grammars is increasingly achieving broad coverage over open-domain texts for a range of constraint-based frameworks (e.g., TAG, LFG, HPSG and CCG), and is being used in real-world applications including information extraction, question answering, grammar checking and machine translation (Uszkoreit, 2002; Oepen et al., 2004; Frank et al., 2006; Zhang and Kordoni, 2008; MacKinlay et al., 2009). In this context, a “precision grammar” is a grammar which has been engineered to model grammaticality, and contrasts with a treebank-induced grammar, for example.

Inevitably, however, such applications demand complete parsing outputs, based on the assumption that the text under investigation will be completely analysable by the grammar. As precision grammars

generally make strong assumptions about complete lexical coverage and grammaticality of the input, their utility is limited over noisy or domain-specific data. This lack of complete coverage can make parsing with precision grammars less attractive than parsing with shallower methods.

One technique that has been successfully applied to improve parser and grammar coverage over a given corpus is error mining (van Noord, 2004; de Kok et al., 2009), whereby  $n$ -grams with low “parsability” are gathered from the large-scale output of a parser as an indication of parser or (precision) grammar errors. However, error mining is very much oriented towards grammar engineering: its results are a mixture of different (mistreated) linguistic phenomena together with engineering errors for the grammar engineer to work through and act upon. Additionally, it generally does not provide any insight into the cause of the parser failure, and it is difficult to identify specific language phenomena from the output.

In this paper, we instead propose a **chart mining** technique that works on intermediate parsing results from a parsing chart. In essence, the method analyses the validity of different analyses for words or constructions based on the “lifetime” and probability of each within the chart, combining the constraints of the grammar with probabilities to evaluate the plausibility of each.

For purposes of exemplification of the proposed technique, we apply chart mining to a deep lexical acquisition (DLA) task, using a maximum entropy-based prediction model trained over a seed lexicon and treebank. The experimental set up is the following: given a set of sentences containing putative instances of English verb particle constructions,

extract a list of non-compositional VPCs optionally with valence information. For comparison, we parse the same sentence set using a state-of-the-art statistical parser, and extract the VPCs from the parser output. Our results show that our chart mining method produces a model which is superior to the treebank parser.

To our knowledge, the only other work that has looked at partial parsing results of precision grammars as a means of linguistic error analysis is that of Kiefer et al. (1999) and Zhang et al. (2007a), where partial parsing models were proposed to select a set of passive edges that together cover the input sequence. Compared to these approaches, our proposed chart mining technique is more general and can be adapted to specific tasks and domains. While we experiment exclusively with an HPSG grammar in this paper, it is important to note that the proposed method can be applied to any grammar formalism which is compatible with chart parsing, and where it is possible to describe an unlexicalised lexical entry for the different categories of lexical item that are to be extracted (see Section 3.2 for details).

The remainder of the paper is organised as follows. Section 2 defines the task of VPC extraction. Section 3 presents the chart mining technique and the feature extraction process for the VPC extraction task. Section 4 evaluates the model performance with comparison to two competitor models over several different measures. Section 5 further discusses the general applicability of chart mining. Finally, Section 6 concludes the paper.

## 2 Verb Particle Constructions

The particular construction type we target for DLA in this paper is English Verb Particle Constructions (henceforth VPCs). 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; Baldwin and Kim, 2009); for the purposes of our dataset, we assume that all particles are prepositional—by far the most common and productive of the three types—and further restrict our attention to single-particle VPCs (i.e., we ignore VPCs such as *get along together*).

One aspect of VPCs that makes them a particularly challenging target for lexical acquisition is that the verb and particle can be non-contiguous (for instance, *hand the paper in* and *battle right on*). This sets them apart from conventional collocations and terminology (cf., Manning and Schütze (1999), Smadja (1993) and McKeown and Radev (2000)) in that they cannot be captured effectively using *n*-grams, due to their variability in the number and type of words potentially interceding between the verb and the particle. Also, while conventional collocations generally take the form of compound nouns or adjective–noun combinations with relatively simple syntactic structure, VPCs occur with a range of valences. Furthermore, VPCs are highly productive in English and vary in use across domains, making them a prime target for lexical acquisition (Dehé, 2002; Baldwin, 2005; Baldwin and Kim, 2009).

In the VPC dataset we use, there is an additional distinction between compositional and non-compositional VPCs. With compositional VPCs, the semantics of the verb and particle both correspond to the semantics of the respective simplex words, including the possibility of the semantics being specific to the VPC construction in the case of particles. For example, *battle on* would be classified as compositional, as the semantics of *battle* is identical to that for the simplex verb, and the semantics of *on* corresponds to the continuative sense of the word as occurs productively in VPCs (cf., *walk/dance/drive/govern/... on*). With non-compositional VPCs, on the other hand, the semantics of the VPC is somehow removed from that of the parts. In the dataset we used for evaluation, we are interested in extracting exclusively non-compositional VPCs, as they require lexicalisation; compositional VPCs can be captured via lexical rules and are hence not the target of extraction.

English VPCs can occur with a number of valences, with the two most prevalent and productive valences being the simple transitive (e.g., *hand in the paper*) and intransitive (e.g., *back off*). For the purposes of our target task, we focus exclusively on these two valence types.

Given the above, we define the English VPC extraction task to be the production of triples of the form  $\langle v, p, s \rangle$ , where  $v$  is a verb lemma,  $p$  is a prepositional particle, and  $s \in \{intrans, trans\}$  is the va-

lence; additionally, each triple has to be semantically non-compositional. The triples are extracted relative to a set of putative token instances for each of the intransitive and transitive valences for a given VPC. That is, a given triple should be classified as positive if and only if it is associated with at least one non-compositional token instance in the provided token-level data.

The dataset used in this research is the one used in the LREC 2008 Multiword Expression Workshop Shared Task (Baldwin, 2008).<sup>1</sup> In the dataset, there is a single file for each of 4,090 candidate VPC triples, containing up to 50 sentences that have the given VPC taken from the British National Corpus. When the valence of the VPC is ignored, the dataset contains 440 unique VPCs among 2,898 VPC candidates. In order to be able to fairly compare our method with a state-of-the-art lexicalised parser trained over the WSJ training sections of the Penn Treebank, we remove any VPC types from the test set which are attested in the WSJ training sections. This removes 696 VPC types from the test set, and makes the task even more difficult, as the remaining testing VPC types are generally less frequent ones. At the same time, it unfortunately means that our results are not directly comparable to those for the original shared task.<sup>2</sup>

### 3 Chart Mining for Parsing with a Large Precision Grammar

#### 3.1 The Technique

The chart mining technique we use in this paper is couched in a constituent-based bottom-up chart parsing paradigm. A parsing chart is a data structure that records all the (complete or incomplete) intermediate parsing results. Every passive edge on the parsing chart represents a complete local analysis covering a sub-string of the input, while each active edge predicts a potential local analysis. In this view, a full analysis is merely a passive edge that spans the whole input and satisfies certain *root con-*

*ditions*. The bottom-up chart parser starts with edges instantiated from lexical entries corresponding to the input words. The grammar rules are used to incrementally create longer edges from smaller ones until no more edges can be added to the chart.

Standardly, the parser returns only outputs that correspond to passive edges in the parsing chart that span the full input string. For those inputs without a full-spanning edge, no output is generated, and the chart becomes the only source of parsing information.

A parsing chart takes the form of a hierarchy of edges. Where only passive edges are concerned, each non-lexical edge corresponds to exactly one grammar rule, and is connected with one or more daughter edge(s), and zero or more parent edge(s). Therefore, traversing the chart is relatively straightforward.

There are two potential challenges for the chart-mining technique. First, there is potentially a huge number of parsing edges in the chart. For instance, when parsing with a large precision grammar like the HPSG English Resource Grammar (ERG, Flickinger (2002)), it is not unusual for a 20-word sentence to receive over 10,000 passive edges. In order to achieve high efficiency in parsing (as well as generation), ambiguity packing is usually used to reduce the number of productive passive edges on the parsing chart (Tomita, 1985). For constraint-based grammar frameworks like LFG and HPSG, subsumption-based packing is used to achieve a higher packing ratio (Oepen and Carroll, 2000), but this might also potentially lead to an inconsistent packed parse forest that does not unpack successfully. For chart mining, this means that not all passive edges are directly accessible from the chart. Some of them are packed into others, and the derivatives of the packed edges are not generated. Because of the ambiguity packing, zero or more local analyses may exist for each passive edge on the chart, and the cross-combination of the packed daughter edges is not guaranteed to be compatible. As a result, expensive unification operations must be reapplied during the unpacking phase. Carroll and Oepen (2005) and Zhang et al. (2007b) have proposed efficient  $k$ -best unpacking algorithms that can selectively extract the most probable readings from the packed parse forest according to a discrimina-

<sup>1</sup>Downloadable from <http://www.csse.unimelb.edu.au/research/lt/resources/vpc/vpc.tgz>.

<sup>2</sup>In practice, there was only one team who participated in the original VPC task (Ramisch et al., 2008), who used a variety of web- and dictionary-based features suited more to high-frequency instances in high-density languages, so a simplistic comparison would not have been meaningful.

tive parse disambiguation model, by minimising the number of potential unifications. The algorithm can be applied to unpack any passive edges. Because of the dynamic programming used in the algorithm and the hierarchical structure of the edges, the cost of the unpacking routine is empirically linear in the number of desired readings, and  $O(1)$  when invoked more than once on the same edge.

The other challenge concerns the selection of informative and representative pieces of knowledge from the massive sea of partial analyses in the parsing chart. How to effectively extract the indicative features for a specific language phenomenon is a very task-specific question, as we will show in the context of the VPC extraction task in Section 3.2. However, general strategies can be applied to generate parse ranking scores on each passive edge. The most widely used parse ranking model is the log-linear model (Abney, 1997; Johnson et al., 1999; Toutanova et al., 2002). When the model does not use non-local features, the accumulated score on a sub-tree under a certain (unpacked) passive edge can be used to approximate the probability of the partial analysis conditioned on the sub-string within that span.<sup>3</sup>

### 3.2 The Application: Acquiring Features for VPC Extraction

As stated above, the target task we use to illustrate the capabilities of our chart mining method is VPC extraction.

The grammar we apply our chart mining method to in this paper is the English Resource Grammar (ERG, Flickinger (2002)), a large-scale precision HPSG for English. Note, however, that the method is equally compatible with any grammar or grammar formalism which is compatible with chart parsing.

The lexicon of the ERG has been semi-automatically extended with VPCs extracted by Baldwin (2005). In order to show the effectiveness of chart mining in discovering “unknowns” and remove any lexical probabilities associated with pre-existing lexical entries, we block the

<sup>3</sup>To have a consistent ranking model on any sub-analysis, one would have to retrain the disambiguation model on every passive edge. In practice, we find this to be intractable. Also, the approximation based on full-parse ranking model works reasonably well.

lexical entries for the verb in the candidate VPC by substituting the input token with a DUMMY-V token, which is coupled with four candidate lexical entries of type: (1) intransitive simplex verb ( $v\_e$ ), (2) transitive simplex verb ( $v\_np\_le$ ), (3) intransitive VPC ( $v\_p\_le$ ), and (4) transitive VPC ( $v\_p\_np\_le$ ), respectively. These four lexical entries represent the two VPC valences we wish to distinguish between in the VPC extraction task, and the competing simplex verb candidates. Based on these lexical types, the features we extract with chart mining are summarised in Table 1. The maximal constituent (MAXCONS) of a lexical entry is defined to be the passive edge that is an ancestor of the lexical entry edge that: (i) must span over the particle, and (ii) has maximal span length. In the case of a tie, the edge with the highest disambiguation score is selected as the MAXCONS. If there is no edge found on the chart that spans over both the verb and the particle, the MAXCONS is set to be NULL, with a MAXSPAN of 0, MAXLEVEL of 0 and MAXCRANK of 4 (see Table 1). The stem of the particle is also collected as a feature.

One important characteristic of these features is that they are completely unlexicalised on the verb. This not only leads to a fair evaluation with the ERG by excluding the influence from the lexical coverage of VPCs in the grammar, but it also demonstrates that complete grammatical coverage over simplex verbs is not a prerequisite for chart mining.

To illustrate how our method works, we present the unpacked parsing chart for the candidate VPC *show off* and input sentence *The boy shows off his new toys* in Figure 1. The non-terminal edges are marked with their syntactic categories, i.e., HPSG rules (e.g., *subjh* for the subject-head-rule, *hadj* for the head-adjunct-rule, etc.), and optionally their disambiguation scores. By traversing upward through parent edges from the DUMMY-V edge, all features can be efficiently extracted (see the third column in Table 1).

It should be noted that none of these features are used to deterministically dictate the predicted VPC category. Instead, the acquired features are used as inputs to a statistical classifier for predicting the type of the VPC candidate at the token level (in the context of the given sentence). In our experiment, we used a maximum entropy-based model to do a 3-

Feature	Description	Examples
LE:MAXCONS	A lexical entry together with the maximal constituent constructed from it	<i>v_-le:subjh</i> , <i>v_np_le:hadj</i> , <i>v_p_le:subjh</i> , <i>v_p_np_le:subj</i>
LE:MAXSPAN	A lexical entry together with the length of the span of the maximal constituent constructed from the LE	<i>v_-le:7</i> , <i>v_np_le:5</i> , <i>v_p_le:4</i> , <i>v_p_np_le:7</i>
LE:MAXLEVEL	A lexical entry together with the levels of projections before it reaches its maximal constituent	<i>v_-le:2</i> , <i>v_np_le:1</i> , <i>v_p_le:2</i> , <i>v_p_np_le:3</i>
LE:MAXCRANK	A lexical entry together with the relative disambiguation score ranking of its maximal constituent among all MaxCons from different LEs	<i>v_-le:4</i> , <i>v_np_le:3</i> , <i>v_p_le:1</i> , <i>v_p_np_le:2</i>
PARTICLE	The stem of the particle in the candidate VPC	<i>off</i>

Table 1: Chart mining features used for VPC extraction

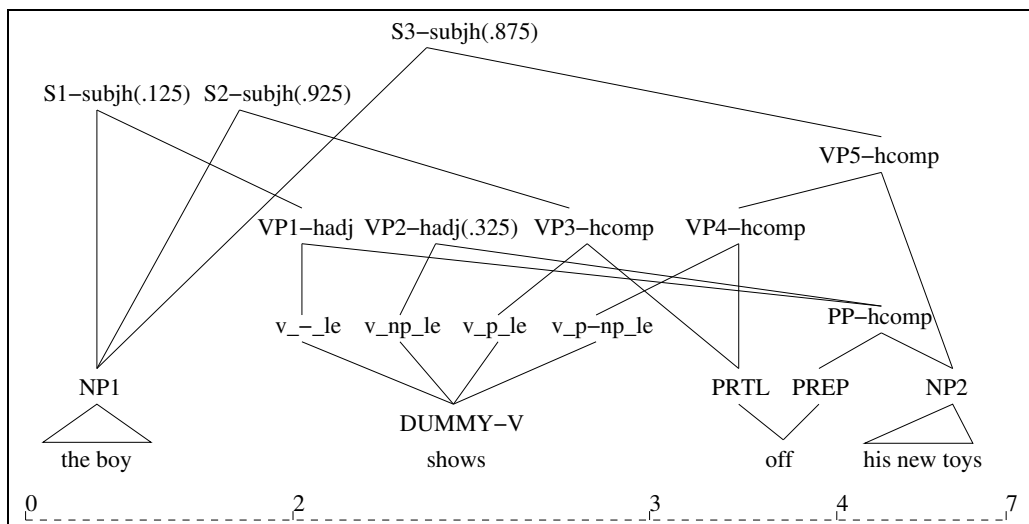


Figure 1: Example of a parsing chart in chart-mining for VPC extraction with the ERG

category classification: non-VPC, transitive VPC, or intransitive VPC. For the parameter estimation of the ME model, we use the TADM open source toolkit (Malouf, 2002). The token-level predictions are then combined with a simple majority voting to derive the type-level prediction for the VPC candidate. In the case of a tie, the method backs off to the naïve baseline model described in Section 4.2, which relies on the combined probability of the verb and particle forming a VPC.

We have also experimented with other ways of deriving type-level predictions from token-level classification results. For instance, we trained a separate classifier that takes the token-level prediction as input in order to determine the type-level VPC predic-

tion. Our results indicate no significant difference between these methods and the basic majority voting approach, so we present results exclusively for this simplistic approach in this paper.

## 4 Evaluation

### 4.1 Experiment Setup

To evaluate the proposed chart mining-based VPC extraction model, we use the dataset from the LREC 2008 Multiword Expression Workshop shared task (see Section 2). We use this dataset to perform three distinct DLA tasks, as detailed in Table 2.

The chart mining feature extraction is implemented as an extension to the PET parser (Callmeier,

Task	Description
GOLD VPC	Determine the valence for a verb–preposition combination which is known to occur as a non-compositional VPC (i.e. known VPC, with unknown valence(s))
FULL	Determine whether each verb–preposition combination is a VPC or not, and further predict its valence(s) (i.e. unknown if VPC, and unknown valence(s))
VPC	Determine whether each verb–preposition combination is a VPC or not <i>ignoring valence</i> (i.e. unknown if VPC, and don’t care about valence)

Table 2: Definitions of the three DLA tasks

2001). We use a slightly modified version of the ERG in our experiments, based on the *nov-06* release. The modifications include 4 newly-added dummy lexical entries for the verb DUMMY-V and the corresponding inflectional rules, and a lexical type prediction model (Zhang and Kordoni, 2006) trained on the LOGON Treebank (Oepen et al., 2004) for unknown word handling. The parse disambiguation model we use is also trained on the LOGON Treebank. Since the parser has no access to any of the verbs under investigation (due to the DUMMY-V substitution), those VPC types attested in the LOGON Treebank do not directly impact on the model’s performance. The chart mining feature extraction process took over 10 CPU days, and collected a total of 44K events for 4,090 candidate VPC triples.<sup>4</sup> 5-fold cross validation is used to train/test the model. As stated above (Section 2), the VPC triples attested in the WSJ training sections of the Penn Treebank are excluded in each testing fold for comparison with the Charniak parser-based model (see Section 4.2).

## 4.2 Baseline and Benchmark

For comparison, we first built a naïve baseline model using the combined probabilities of the verb and particle being part of a VPC. More specifically,  $P(c|v)$  and  $P(c|p)$  are the probabilities of a given verb  $v$  and particle  $p$  being part of a VPC candidate of type  $s \in \{intrans, trans, null\}$ , for transitive

<sup>4</sup>Not all sentences in the dataset are successfully chart-mined. Due to the complexity of the precision grammar we use, the parser is unlikely to complete the parsing chart for extremely long sentences (over 50 words). Moreover, sentences which do not receive any spanning edge over the verb and the particle are not considered as an indicative event. Nevertheless, the coverage of the chart mining is much higher than the full-parse coverage of the grammar.

VPC, intransitive VPC, and non-VPC, respectively.  $\tilde{P}(s|v, p) = P(s|v) \cdot P(s|p)$  is used to approximate the joint probability of verb-particle  $(v, p)$  being of type  $s$ , and the prediction type is chosen randomly based on this probabilistic distribution. Both  $P(s|v)$  and  $P(s|p)$  can be estimated from a list of VPC candidate types. If  $v$  is unseen,  $P(s|v)$  is set to be  $\frac{1}{|V|} \sum_{v_i \in V} P(s|v_i)$  estimated over all verbs  $|V|$  seen in the list of VPC candidates. The naïve baseline performed poorly, mainly because there is not enough knowledge about the context of use of VPCs. This also indicates that the task of VPC extraction is non-trivial, and that context (evidence from sentences in which the VPC putatively occurs) must be incorporated in order to make more accurate predictions.

As a benchmark VPC extraction system, we use the Charniak parser (Charniak, 2000). This statistical parser induces a context-free grammar and a generative parsing model from a training set of gold standard parse trees. Traditionally, it has been trained over the WSJ component of the Penn Treebank, and for this work we decided to take the same approach and train over sections 1 to 22, and use section 23 for parameter-tuning. After parsing, we simply search for the VPC triples in each token instance with `tgrep2`,<sup>5</sup> and decide on the classification of the candidate by majority voting over all instances, breaking ties randomly.

<sup>5</sup>Noting that the Penn POS tagset captures essentially the compositional vs. non-compositional VPC distinction required in the extraction task, through the use of the RP (prepositional particle, for non-compositional VPCs) and RB (adverb, for compositional VPCs) tags.

### 4.3 Results

The results of our experiments are summarised in Table 3. For the naïve baseline and the chart mining-based models, the results are averaged over 5-fold cross validation.

We evaluate the methods in the form of the three tasks described in Table 2. Formally, GOLD VPC equates to extracting  $\langle v, p, s \rangle$  tuples from the subset of gold-standard  $\langle v, p \rangle$  tuples; FULL equates to extracting  $\langle v, p, s \rangle$  tuples for all VPC candidates; and VPC equates to extracting  $\langle v, p \rangle$  tuples (ignoring valence) over all VPC candidates. In each case, we present the precision (P), recall (R) and F-score ( $\beta = 1$ : F). For multi-category classifications (i.e. the two tasks where we predict the valence  $s$ , indicated as “All” in Table 3), we micro-average the precision and recall over the two VPC categories, and calculate the F-score as their harmonic mean.

From the results, it is obvious that the chart mining-based model performs best overall, and indeed for most of the measures presented. The Charniak parser-based extraction method performs reasonably well, especially in the VPC+valence extraction task over the FULL task, where the recall was higher than the chart mining method. Although not reported here, we observe a marked improvement in the results for the Charniak parser when the VPC types attested in the WSJ are not filtered from the test set. This indicates that the statistical parser relies heavily on lexicalised VPC information, while the chart mining model is much more syntax-oriented. In error analysis of the data, we observed that the Charniak parser was noticeably more accurate at extracting VPCs where the verb was frequent (our method, of course, did not have access to the base frequency of the simplex verb), underlining again the power of lexicalisation. This points to two possibilities: (1) the potential for our method to similarly benefit from lexicalisation if we were to remove the constraint on ignoring any pre-existing lexical entries for the verb; and (2) the possibility for hybridising between lexicalised models for frequent verbs and unlexicalised models for infrequent verbs. Having said this, it is important to reinforce that lexical acquisition is usually performed in the absence of lexicalised probabilities, as if we have prior knowledge of the lexical item, there is no need

to extract it. In this sense, the first set of results in Table 3 over Gold VPCs are the most informative, and illustrate the potential of the proposed approach.

From the results of all the models, it would appear that intransitive VPCs are more difficult to extract than transitive VPCs. This is partly because the dataset we use is unbalanced: the number of transitive VPC types is about twice the number of intransitive VPCs. Also, the much lower numbers over the FULL set compared to the GOLD VPC set are due to the fact that only 1/8 of the candidates are true VPCs.

### 5 Discussion and Future Work

The inventory of features we propose for VPC extraction is just one illustration of how partial parse results can be used in lexical acquisition tasks. The general chart mining technique can easily be adapted to learn other challenging linguistic phenomena, such as the countability of nouns (Baldwin and Bond, 2003), subcategorization properties of verbs or nouns (Korhonen, 2002), and general multiword expression (MWE) extraction (Baldwin and Kim, 2009). With MWE extraction, e.g., even though some MWEs are fixed and have no internal syntactic variability, such as *ad hoc*, there is a very large proportion of idioms that allow various degrees of internal variability, and with a variable number of elements. For example, the idiom *spill the beans* allows internal modification (*spill mountains of beans*), passivisation (*The beans were spilled in the latest edition of the report*), topicalisation (*The beans, the opposition spilled*), and so forth (Sag et al., 2002). In general, however, the exact degree of variability of an idiom is difficult to predict (Riehemann, 2001). The chart mining technique we propose here, which makes use of partial parse results, may facilitate the automatic recognition task of even more flexible idioms, based on the encouraging results for VPCs.

The main advantage, though, of chart mining is that parsing with precision grammars does not any longer have to assume complete coverage, as has traditionally been the case. As an immediate consequence, the possibility of applying our chart mining technique to evolving medium-sized grammars makes it especially interesting for lexical acquisi-

Task	VPC Type	Naïve Baseline			Charniak Parser			Chart-Mining		
		P	R	F	P	R	F	P	R	F
GOLD VPC	Intrans-VPC	0.300	0.018	0.034	0.549	0.753	0.635	0.845	0.621	0.716
	Trans-VPC	0.676	0.348	0.459	0.829	0.648	0.728	0.877	0.956	0.915
	All	0.576	0.236	0.335	0.691	0.686	0.688	0.875	0.859	<b>0.867</b>
FULL	Intrans-VPC	0.060	0.018	0.028	0.102	0.593	0.174	0.153	0.155	0.154
	Trans-VPC	0.083	0.348	0.134	0.179	0.448	0.256	0.179	0.362	0.240
	All	0.080	0.236	0.119	0.136	0.500	0.213	0.171	0.298	<b>0.218</b>
	VPC	0.123	0.348	0.182	0.173	0.782	0.284	0.259	0.332	<b>0.291</b>

Table 3: Results for the different methods over the three VPC extraction tasks detailed in Table 2

tion over low-density languages, for instance, where there is a real need for rapid-prototyping of language resources.

The chart mining approach we propose in this paper is couched in the bottom-up chart parsing paradigm, based exclusively on passive edges. As future work, we would also like to look into the top-level active edges (those active edges that are never completed), as an indication of failed assumptions. Moreover, it would be interesting to investigate the applicability of the technique in other parsing strategies, e.g., head-corner or left-corner parsing. Finally, it would also be interesting to investigate whether by using the features we acquire from chart mining enhanced with information on the prevalence of certain patterns, we could achieve performance improvements over broader-coverage tree-bank parsers such as the Charniak parser.

## 6 Conclusion

We have proposed a chart mining technique for lexical acquisition based on partial parsing with precision grammars. We applied the proposed method to the task of extracting English verb particle constructions from a prescribed set of corpus instances. Our results showed that simple unlexicalised features mined from the chart can be used to effectively extract VPCs, and that the model outperforms a probabilistic baseline and the Charniak parser at VPC extraction.

## Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Re-

search Council through the ICT Centre of Excellence program. The first was supported by the German Excellence Cluster of Multimodal Computing and Interaction.

## References

- Steven Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23:597–618.
- Timothy Baldwin and Francis Bond. 2003. Learning the countability of English nouns from corpus data. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 463–470, Sapporo, Japan.
- Timothy Baldwin and Su Nam Kim. 2009. Multiword expressions. In Nitin Indurkha and Fred J. Damerau, editors, *Handbook of Natural Language Processing*. CRC Press, Boca Raton, USA, 2nd edition.
- Timothy Baldwin. 2005. The deep lexical acquisition of English verb-particle constructions. *Computer Speech and Language, Special Issue on Multiword Expressions*, 19(4):398–414.
- Timothy Baldwin. 2008. A resource for evaluating the deep lexical acquisition of English verb-particle constructions. In *Proceedings of the LREC 2008 Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 1–2, Marrakech, Morocco.
- Ulrich Callmeier. 2001. Efficient parsing with large-scale unification grammars. Master’s thesis, Universität des Saarlandes, Saarbrücken, Germany.
- John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*, pages 165–176, Jeju Island, Korea.
- Eugene Charniak. 2000. A maximum entropy-based parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL2000)*, Seattle, USA.
- Daniel de Kok, Jianqiang Ma, and Gertjan van Noord. 2009. A generalized method for iterative error mining in parsing results. In *Proceedings of the ACL2009 Workshop on Grammar Engineering Across Frameworks (GEAF)*, Singapore.



- Nicole Dehé. 2002. *Particle Verbs in English: Syntax, Information, Structure and Intonation*. John Benjamins, Amsterdam, Netherlands/Philadelphia, USA.
- Dan Flickinger. 2002. On building a more efficient grammar by exploiting types. In Stephan Oepen, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*, pages 1–17. CSLI Publications.
- Anette Frank, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crismann, Brigitte Jörg, and Ulrich Schäfer. 2006. Question answering from structured knowledge sources. *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives.*, 5(1):20–48.
- Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, UK.
- Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic unification-based grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999)*, pages 535–541, Maryland, USA.
- Bernd Kiefer, Hans-Ulrich Krieger, John Carroll, and Rob Malouf. 1999. A Bag of Useful Techniques for Efficient and Robust Parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Maryland, USA.
- Anna Korhonen. 2002. *Subcategorization Acquisition*. Ph.D. thesis, University of Cambridge.
- Andrew MacKinlay, David Martinez, and Timothy Baldwin. 2009. Biomedical event annotation with CRFs and precision grammars. In *Proceedings of BioNLP 2009: Shared Task*, pages 77–85, Boulder, USA.
- Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL 2002)*, pages 49–55, Taipei, Taiwan.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.
- Kathleen R. McKeown and Dragomir R. Radev. 2000. Collocations. In Robert Dale, Hermann Moisl, and Harold Somers, editors, *Handbook of Natural Language Processing*.
- Stephan Oepen and John Carroll. 2000. Ambiguity packing in constraint-based parsing — practical results. In *Proceedings of the 1st Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL 2000)*, pages 162–169, Seattle, USA.
- Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. 2004. Som å kapp-ete med trollet? Towards MRS-Based Norwegian–English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore, USA.
- Carlos Ramisch, Paulo Schreiner, Marco Idiart, and Aline Villavicencio. 2008. An evaluation of methods for the extraction of multiword expressions. In *Proceedings of the LREC 2008 Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 50–53, Marrakech, Morocco.
- Susanne Riehemann. 2001. *A Constructional Approach to Idioms and Word Formation*. Ph.D. thesis, Stanford University, CA, USA.
- Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan 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)*, pages 1–15, Mexico City, Mexico.
- Frank Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–178.
- Masaru Tomita. 1985. An efficient context-free parsing algorithm for natural languages. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 756–764, Los Angeles, USA.
- Kristina Toutanova, Christopher D. Manning, Stuart M. Shieber, Dan Flickinger, and Stephan Oepen. 2002. Parse ranking for a rich HPSG grammar. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT 2002)*, pages 253–263, Sozopol, Bulgaria.
- Hans Uszkoreit. 2002. New chances for deep linguistic processing. In *Proceedings of the 19th international conference on computational linguistics (COLING 2002)*, Taipei, Taiwan.
- Gertjan van Noord. 2004. Error mining for wide-coverage grammar engineering. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 446–453, Barcelona, Spain.
- Aline Villavicencio and Ann 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, Korea.
- Yi Zhang and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open texts processing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 275–280, Genoa, Italy.
- Yi Zhang and Valia Kordoni. 2008. Robust parsing with a large HPSG grammar. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.
- Yi Zhang, Valia Kordoni, and Erin Fitzgerald. 2007a. Partial parse selection for robust deep processing. In *Proceedings of ACL 2007 Workshop on Deep Linguistic Processing*, pages 128–135, Prague, Czech Republic.
- Yi Zhang, Stephan Oepen, and John Carroll. 2007b. Efficiency in unification-based N-best parsing. In *Proceedings of the 10th International Conference on Parsing Technologies (IWPT 2007)*, pages 48–59, Prague, Czech.