An Expert Lexicon Approach to Identifying English Phrasal Verbs

Wei Li, Xiuhong Zhang, Cheng Niu, Yuankai Jiang, Rohini Srihari

Cymfony Inc.
600 Essjay Road
Williamsville, NY 14221, USA
{wei, xzhang, cniu, yjiang, rohini}@Cymfony.com

Abstract

Phrasal Verbs are an important feature of the English language. Properly identifying them provides the basis for an English parser to decode the related structures. Phrasal verbs have been a challenge to Natural Language Processing (NLP) because they sit at the borderline between lexicon and syntax. Traditional NLP frameworks that separate the lexicon module from the parser make it difficult to handle this problem properly. This paper presents a finite state approach that integrates a phrasal verb expert lexicon between shallow parsing and deep parsing to handle morpho-syntactic interaction. With precision/recall combined performance benchmarked consistently at 95.8%-97.5%, the Phrasal Verb identification problem has basically been solved with the presented method.

1 Introduction

Any natural language processing (NLP) system needs to address the issue of handling multiword expressions, including Phrasal Verbs (PV) [Sag *et al.* 2002; Breidt *et al.* 1996]. This paper presents a proven approach to identifying English PVs based on pattern matching using a formalism called Expert Lexicon.

Phrasal Verbs are an important feature of the English language since they form about one third of the English verb vocabulary. [1] Properly recognizing PVs is an important condition for English parsing. Like single-word verbs, each PV has its own lexical features including subcategorization features that determine its structural patterns [Fraser 1976; Bolinger 1971; Pelli 1976; Shaked 1994], e.g., *look for* has syntactic subcategorization and semantic features similar to those of *search*; *carry…on* shares lexical features with *continue*. Such lexical features can be represented in the PV lexicon in the same way as those for single-word verbs, but a parser can only use them when the PV is identified.

Problems like PVs are regarded as 'a pain in the neck for NLP' [Sag *et al.* 2002]. A proper solution to this problem requires tighter interaction between syntax and lexicon than traditionally available [Breidt *et al.* 1994]. Simple lexical lookup leads to severe degradation in both precision and recall, as our benchmarks show (Section 4). The recall problem is mainly due to separable PVs such as *turn…off* which allow for syntactic units to be inserted inside the PV compound, e.g., *turn it off*, *turn the radio off*. The precision problem is caused by the ambiguous function of the particle. For example, a simple lexical lookup will mistag *looked for* as a phrasal verb in sentences such as *He looked for quite a while but saw nothing*.

In short, the traditional NLP framework that separates the lexicon module from a parser makes it difficult to handle this problem properly. This paper presents an expert lexicon approach that integrates the lexical module with contextual checking based on shallow parsing results. Extensive blind benchmarking shows that this approach is very effective for identifying phrasal verbs, resulting in the precision/recall combined F-score of about 96%.

The remaining text is structured as follows. Section 2 presents the problem and defines the task. Section 3 presents the Expert Lexicon

[1] For the verb vocabulary of our system based on machine-readable dictionaries and two Phrasal Verb dictionaries, phrasal verb entries constitute 33.8% of the entries.

formalism and illustrates the use of this formalism in solving this problem. Section 4 shows the benchmarking and analysis, followed by conclusions in Section 5.

2 Phrasal Verb Challenges

This section defines the problems we intend to solve, with a checklist of tasks to accomplish.

2.1 Task Definition

First, we define the task as the *identification* of PVs in support of deep parsing, not as the *parsing* of the structures headed by a PV. These two are separated as two tasks not only because of modularity considerations, but more importantly based on a natural labor division between NLP modules.

Essential to the second argument is that these two tasks are of a different linguistic nature: the identification task belongs to (compounding) morphology (although it involves a syntactic interface) while the parsing task belongs to syntax. The naturalness of this division is reflected in the fact that there is no need for a specialized, PV-oriented parser. The same parser, mainly driven by lexical subcategorization features, can handle the structural problems for both phrasal verbs and other verbs. The following active and passive structures involving the PVs *look after* (corresponding to *watch*) and *carry…on* (corresponding to *continue*) are decoded by our deep parser after PV identification: *she is being carefully 'looked after' (watched); we should 'carry on' (continue) the business for a while*.

There has been no unified definition of PVs among linguists. Semantic compositionality is often used as a criterion to distinguish a PV from a syntactic combination between a verb and its associated adverb or prepositional phrase [Shaked 1994]. In reality, however, PVs reside in a continuum from opaque to transparent in terms of semantic compositionality [Bolinger 1971]. There exist fuzzy cases such as *take something away*[2] that may be included either as a PV or as a regular syntactic sequence. There is agreement on the vocabulary scope for the majority of PVs, as reflected in the overlapping of PV entries from major English dictionaries.

English PVs are generally classified into three major types. Type I usually takes the form of an intransitive verb plus a particle word that originates from a preposition. Hence the resulting compound verb has become transitive, e.g., *look for, look after, look forward to, look into,* etc.

Type II typically takes the form of a transitive verb plus a particle from the set {on, off, up, down}, e.g., *turn…on, take…off, wake…up, let…down*. Marginal cases of particles may also include {out, in, away} such as *take…away, kick …in, pull…out*.[3]

Type III takes the form of an intransitive verb plus an adverb particle, e.g., *get by, blow up, burn up, get off,* etc. Note that Type II and Type III PVs have considerable overlapping in vocabulary, e.g., *The bomb blew up* vs. *The clown blew up the balloon*. The overlapping phenomenon can be handled by assigning both a transitive feature and an intransitive feature to the identified PVs in the same way that we treat the overlapping of single-word verbs.

The first issue in handling PVs is inflection. A system for identifying PVs should match the inflected forms, both regular and irregular, of the leading verb.

The second is the representation of the lexical identity of recognized PVs. This is to establish a PV (a compound word) as a syntactic atomic unit with all its lexical properties determined by the lexicon [Di Sciullo and Williams 1987]. The output of the identification module based on a PV lexicon should support syntactic analysis and further processing. This translates into two sub-tasks: (i) lexical feature assignment, and (ii) canonical form representation. After a PV is identified, its lexical features encoded in the PV lexicon should be assigned for a parser to use. The representation of a canonical form for an identified PV is necessary to allow for individual rules to be associated with identified PVs in further processing and to facilitate verb retrieval in applications. For example, if we use *turn_off* as the canonical form for the PV *turn…off*, identified in both *he turned off the radio* and *he*

[2] Single-word verbs like 'take' are often over-burdened with dozens of senses/uses. Treating marginal cases like 'take…away' as independent phrasal verb entries has practical benefits in relieving the burden and the associated noise involving 'take'.

[3] These three are arguably in the gray area. Since they do not fundamentally affect the meaning of the leading verb, we do not have to treat them as phrasal verbs. In principle, they can also be treated as adverb complements of verbs.

turned the radio off, a search for *turn_off* will match all and only the mentions of this PV.

The fact that PVs are separable hurts recall. In particular, for Type II, a Noun Phrase (NP) object can be inserted inside the compound verb. NP insertion is an intriguing linguistic phenomenon involving the morpho-syntactic interface: a morphological compounding process needs to interact with the formation of a syntactic unit.

Type I PVs also have the separability problem, albeit to a lesser degree. The possible inserted units are adverbs in this case, e.g., *look **everywhere** for, look **carefully** after*.

What hurts precision is spurious matches of PV negative instances. In a sentence with the structure V+[P+NP], [V+P] may be mistagged as a PV, as seen in the following pairs of examples for Type I and Type II:

(1a) She [**looked for**] you yesterday.
(1b) She **looked** [**for** quite a while] (but saw nothing).
(2a) She [**put on**] the coat.
(2b) She **put** [**on** the table] the book she borrowed yesterday.

To summarize, the following is a checklist of problems that a PV identification system should handle: (i) verb inflection, (ii) lexical identity representation, (iii) separability, and (iv) negative instances.

2.2 Related Work

Two lines of research are reported in addressing the PV problem: (i) the use of a high-level grammar formalism that integrates the identification with parsing, and (ii) the use of a finite state device in identifying PVs as a lexical support for the subsequent parser. Both approaches have their own ways of handling the morpho-syntactic interface.

[Sag *et al.* 2002] and [Villavicencio *et al.* 2002] present their project *LinGO-ERG* that handles PV identification and parsing together. LingGO-ERG is based on Head-driven Phrase Structure Grammar (HPSG), a unification-based grammar formalism. HPSG provides a mono-stratal lexicalist framework that facilitates handling intricate morpho-syntactic interaction. PV-related morphological and syntactic structures are accounted for by means of a lexical selection mechanism where the verb morpheme subcategorizes for its syntactic object in addition to its particle morpheme.

The LingGO-ERG lexicalist approach is believed to be effective. However, their coverage and testing of the PVs seem preliminary. The LinGO-ERG lexicon contains 295 PV entries, with no report on benchmarks.

In terms of the restricted flexibility and modifiability of a system, the use of high-level grammar formalisms such as HPSG to integrate identification in deep parsing cannot be compared with the alternative finite state approach [Breidt *et al.* 1994].

[Breidt *et al.*1994]'s approach is similar to our work. Multiword expressions including idioms, collocations, and compounds as well as PVs are accounted for by using *local grammar* rules formulated as regular expressions. There is no detailed description for English PV treatment since their work focuses on multilingual, multi-word expressions in general. The authors believe that the local grammar implementation of multiword expressions can work with general syntax either implemented in a high-level grammar formalism or implemented as a local grammar for the required morpho-syntactic interaction, but this interaction is not implemented into an integrated system and hence it is impossible to properly measure performance benchmarks.

There is no report on implemented solutions covering the entire English PVs that are fully integrated into an NLP system and are well tested on sizable real life corpora, as is presented in this paper.

3 Expert Lexicon Approach

This section illustrates the system architecture and presents the underlying Expert Lexicon (EL) formalism, followed by the description of the implementation details.

3.1 System Architecture

Figure 1 shows the system architecture that contains the PV Identification Module based on the PV Expert Lexicon.

This is a pipeline system mainly based on pattern matching implemented in local grammars and/or expert lexicons [Srihari *et al* 2003].[4]

[4] POS and NE tagging are hybrid systems involving both hand-crafted rules and statistical learning.

English parsing is divided into two tasks: shallow parsing and deep parsing. The shallow parser constructs Verb Groups (VGs) and basic Noun Phrases (NPs), also called *BaseNPs* [Church 1988]. The deep parser utilizes syntactic subcategorization features and semantic features of a head (e.g., VG) to decode both syntactic and logical dependency relationships such as Verb-Subject, Verb-Object, Head-Modifier, etc.

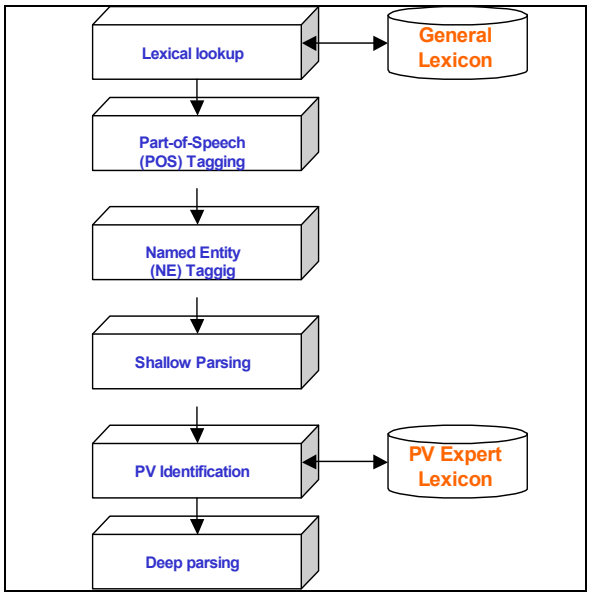

Figure 1. System Architecture

The general lexicon lookup component involves stemming that transforms regular or irregular inflected verbs into the base forms to facilitate the later phrasal verb matching. This component also performs indexing of the word occurrences in the processed document for subsequent expert lexicons.

The PV Identification Module is placed between the Shallow Parser and the Deep Parser. It requires shallow parsing support for the required syntactic interaction and the PV output provides lexical support for deep parsing.

Results after shallow parsing form a proper basis for PV identification. First, the inserted NPs and adverbial time NEs are already constructed by the shallow parser and NE tagger. This makes it easy to write pattern matching rules for identifying separable PVs.

Second, the constructed basic units NE, NP and VG provide conditions for constraint-checking in PV identification. For example, to prevent spurious matches in sentences like *she put the coat on the table,* it is necessary to check that the post-particle unit should NOT be an NP. The VG chunking also decodes the voice, tense and aspect features that can be used as additional constraints for PV identification. A sample macro rule *active_V_Pin* that checks the 'NOT passive' constraint and the 'NOT time', 'NOT location' constraints is shown in 3.3.

3.2 Expert Lexicon Formalism

The Expert Lexicon used in our system is an index-based formalism that can associate pattern matching rules with lexical entries. It is organized like a lexicon, but has the power of a lexicalized local grammar.

All Expert Lexicon entries are indexed, similar to the case for the finite state tool in INTEX [Silberztein 2000]. The pattern matching time is therefore reduced dramatically compared to a sequential finite state device [Srihari *et al.* 2003].[5]

The expert lexicon formalism is designed to enhance the lexicalization of our system, in accordance with the general trend of lexicalist approaches to NLP. It is especially beneficial in handling problems like PVs and many individual or idiosyncratic linguistic phenomena that can not be covered by non-lexical approaches.

Unlike the extreme lexicalized word expert system in [Small and Rieger 1982] and similar to the IDAREX local grammar formalism [Breidt *et al.*1994], our EL formalism supports a parameterized macro mechanism that can be used to capture the general rules shared by a set of individual entries. This is a particular useful mechanism that will save time for computational lexicographers in developing expert lexicons, especially for phrasal verbs, as shall be shown in Section 3.3 below.

The Expert Lexicon tool provides a flexible interface for coordinating lexicons and syntax: any number of expert lexicons can be placed at any levels, hand-in-hand with other non-lexicalized modules in the pipeline architecture of our system.

[5] Some other unique features of our EL formalism include: (i) providing the capability of proximity checking as rule constraints in addition to pattern matching using regular expressions so that the rule writer or lexicographer can exploit the combined advantages of both, and (ii) the propagation functionality of semantic tagging results, to accommodate principles like *one sense per discourse*.

3.3 Phrasal Verb Expert Lexicon

To cover the three major types of PVs, we use the macro mechanism to capture the shared patterns. For example, the NP insertion for Type II PV is handled through a macro called V_NP_P, formulated in pseudo code as follows.

```
V_NP_P($V,$P,$V_P,$F1, $F2,…) :=
    Pattern:
        $V
        NP
        ('right'|'back'|'straight')
        $P
        NOT NP
    Action:
        $V: %assign_feature($F1, $F2,…)
            %assign_canonical_form($V_P)
        $P: %deactivate
```

This macro represents cases like ***Take*** *the coat* ***off****, please;* ***put*** *it back* ***on****, it's raining now*. It consists of two parts: 'Pattern' in regular expression form (with parentheses for optionality, a bar for logical OR, a quoted string for checking a word or head word) and 'Action' (signified by the prefix %). The parameters used in the macro (marked by the prefix $) include the leading verb $V, particle $P, the canonical form $V_P, and features F_n$. After the defined pattern is matched, a Type II separable verb is identified. The *Action* part ensures that the lexical identity be represented properly, i.e. the assignment of the lexical features and the canonical form. The *deactivate* action flags the particle as being part of the phrasal verb.

In addition, to prevent a spurious case in (3b), the macro V_NP_P checks the contextual constraints that no NP (i.e. NOT NP) should follow a PV particle. In our shallow parsing, NP chunking does not include identified time NEs, so it will not block the PV identification in (3c).

(3a) She [**put** the coat **on**].
(3b) She **put** the coat [**on** the table].
(3c) She [**put** the coat **on**] yesterday.

All three types of PVs when used without NP insertion are handled by the same set of macros, due to the formal patterns they share. We use a set of macros instead of one single macro, depending on the type of particle and the voice of the verb, e.g., *look for* calls the macro [active_V_Pfor | passive_V_Pfor], *fly in* calls the macro [active_V_Pin | passive_V_Pin], etc.

The distinction between active rules and passive rules lies in the need for different constraints. For example, a passive rule needs to check the post-particle constraint [NOT NP] to block the spurious case in (4b).

(4a) He [**turned on**] the radio.
(4b) The world [had been **turned**] [**on** its head] again.

As for particles, they also require different constraints in order to block spurious matches. For example, *active_V_Pin* (formulated below) requires the constraints 'NOT location NOT time' after the particle while *active_V_Pfor* only needs to check 'NOT time', shown in (5) and (6).

(5a) Howard [had **flown in**] from Atlanta.
(5b) The rocket [would **fly**] [**in** 1999].
(6a) She was [**looking for]** California on the map.
(6b) She **looked** [**for** quite a while].

```
active_V_Pin($V, in, $V_P,$F1, $F2,…) :=
Pattern:
    $V NOT passive
    (Adv|time)
    $P
    NOT location NOT time
Action:
    $V: %assign_feature($F1, $F2, …)
        %assign_canonical_form($V_P)
     $P: %deactivate
```

The coding of the few PV macros requires skilled computational grammarians and a representative development corpus for rule debugging. In our case, it was approximately 15 person-days of skilled labor including data analysis, macro formulation and five iterations of debugging against the development corpus. But after the PV macros are defined, lexicographers can quickly develop the PV entries: it only cost one person-day to enter the entire PV vocabulary using the EL formalism and the implemented macros. We used the *Cambridge International Dictionary of Phrasal Verbs* and *Collins Cobuild Dictionary of Phrasal Verbs* as the major reference for developing our PV Expert

Lexicon.[6] This expert lexicon contains 2,590 entries. The EL-rules are ordered with specific rules placed before more general rules. A sample of the developed PV Expert Lexicon is shown below (the prefix @ denotes a macro call):

```
abide:  @V_P_by(abide, by, abide_by, V6A,
        APPROVING_AGREEING)
accede: @V_P_to(accede, to, accede_to, V6A,
        APPROVING_AGREEING)
add:    @V_P(add, up, add_up, V2A,
        MATH_REASONING);
        @V_NP_P(add, up, add_up, V6A,
        MATH_REASONING)
        ……
```

In the above entries, V6A and V2A are subcategorization features for transitive and intransitive verb respectively, while APPROVING_AGREEING and MATH_REASONING are semantic features. These features provide the lexical basis for the subsequent parser.

The PV identification method as described above resolves all the problems in the checklist. The following sample output shows the identification result:

```
NP[That]
VG[could slow: slow_down/V6A/MOVING]
NP[him]
down/deactivated .
```

4 Benchmarking

Blind benchmarking was done by two non-developer testers manually checking the results. In cases of disagreement, a third tester was involved in examining the case to help resolve it. We ran benchmarking on both the formal style and informal style of English text.

4.1 Corpus Preparation

Our development corpus (around 500 KB) consists of the MUC-7 (Message Understanding Conference-7) *dryrun* corpus and an additional collection of news domain articles from TREC (Text Retrieval Conference) data. The PV expert lexicon rules, mainly the macros, were developed and debugged using the development corpus.

The first testing corpus (called English-zone corpus) was downloaded from a website that is designed to teach PV usage in Colloquial English (http://www.english-zone.com/phrasals/w-phrasals.html). It consists of 357 lines of sample sentences containing 347 PVs. This addresses the sparseness problem for the less frequently used PVs that rarely get benchmarked in running text testing. This is a concentrated corpus involving varieties of PVs from text sources of an informal style, as shown below.[7]

"Would you *care for* some dessert? We have ice cream, cookies, or cake."
Why are you *wrapped up* in that blanket?
After John's wife died, he had to *get through* his sadness.
After my sister cut her hair by herself, we had to take her to a hairdresser to *even* her hair *out*!
After the fire, the family had to *get by* without a house.

We have prepared two collections from the running text data to test written English of a more formal style in the general news domain: (i) the MUC-7 *formal run* corpus (342 KB) consisting of 99 news articles, and (ii) a collection of 23,557 news articles (105MB) from the TREC data.

4.2 Performance Testing

There is no available system known to the NLP community that claims a capability for PV treatment and could thus be used for a reasonable performance comparison. Hence, we have devised a *bottom-line* system and a *baseline* system for comparison with our EL-driven system. The bottom-line system is defined as a simple lexical lookup procedure enhanced with the ability to match inflected verb forms but with no capability of checking contextual constraints. There is no discussion in the literature on what

[6] Some entries that are listed in these dictionaries do not seem to belong to phrasal verb categories, e.g., *relieve…of* (as used in *relieve somebody of something*), *remind…of* (as used in *remind somebody of something*), etc. It is generally agreed that such cases belong to syntactic patterns in the form of V+NP+P+NP that can be captured by subcategorization. We have excluded these cases.

[7] Proper treatment of PVs is most important in parsing text sources involving Colloquial English, e.g., interviews, speech transcripts, chat room archives. There is an increasing demand for NLP applications in handling this type of data.

constitutes a reasonable baseline system for PV. We believe that a baseline system should have the additional, easy-to-implement ability to jump over inserted object case pronouns (e.g., *turn **it** on*) and adverbs (e.g., *look **everywhere** for*) in PV identification.

Both the MUC-7 formal run corpus and the English-zone corpus were fed into the bottom-line and the baseline systems as well as our EL-driven system described in Section 3.3. The benchmarking results are shown in Table 1 and Table 2. The F-score is a combined measure of precision and recall, reflecting the overall performance of a system.

Table 1. Running Text Benchmarking 1

	Bottom-line	Baseline	EL
Correct	303	334	338
Missing	58	27	23
Spurious	33	34	7
Precision	90.2%	88.4%	98.0%
Recall	83.9%	92.5%	93.6%
F-score	**86.9%**	**91.6%**	**95.8%**

Table 2. Sampling Corpus Benchmarking

	Bottom-line	Baseline	EL
Correct	215	244	324
Missing	132	103	23
Spurious	0	0	0
Precision	100%	100%	100%
Recall	62.0%	70.3%	93.4%
F-score	**76.5%**	**82.6%**	**96.6%**

Compared with the bottom-line performance and the baseline performance, the F-score for the presented method has surged 9-20 percentage points and 4-14 percentage points, respectively.

The high precision (100%) in Table 2 is due to the fact that, unlike running text, the sampling corpus contains only positive instances of PV. This weakness, often associated with sampling corpora, is overcome by benchmarking running text corpora (Table 1 and Table 3).

To compensate for the limited size of the MUC formal run corpus, we used the testing corpus from the TREC data. For such a large testing corpus (23,557 articles, 105MB), it is impractical for testers to read every article to count mentions of all PVs in benchmarking. Therefore, we selected three representative PVs *look for, turn…on* and *blow…up* and used the head verbs (*look, turn, blow*), including their inflected forms, to retrieve all sentences that contain those verbs. We then ran the retrieved sentences through our system for benchmarking (Table 3).

All three of the blind tests show fairly consistent benchmarking results (F-score 95.8%-97.5%), indicating that these benchmarks reflect the true capability of the presented system, which targets the entire PV vocabulary instead of a selected subset. Although there is still some room for further enhancement (to be discussed shortly), the PV identification problem is basically solved.

Table 3. Running Text Benchmarking 2

	'look for'	'turn…on'	'blow…up'
Correct	1138	128	650
Missing	76	0	33
Spurious	5	9	0
Precision	99.6%	93.4%	100.0%
Recall	93.7%	100.0%	95.2%
F-score	**96.6%**	**97.5%**	**97.5%**

4.3 Error Analysis

There are two major factors that cause errors: (i) the impact of errors from the preceding modules (POS and Shallow Parsing), and (ii) the mistakes caused by the PV Expert Lexicon itself.

The POS errors caused more problems than the NP grouping errors because the inserted NP tends to be very short, posing little challenge to the BaseNP shallow parsing. Some verbs mis-tagged as nouns by POS were missed in PV identification.

There are two problems that require the fine-tuning of the PV Identification Module. First, the macros need further adjustment in their constraints. Some constraints seem to be too strong or too weak. For example, in the Type I macro, although we expected the possible insertion of an adverb, however, the constraint on allowing for only one optional adverb and not allowing for a time adverbial is still too strong. As a result, the system failed to identify *listening…to* and *meet…with* in the following cases: *…was not listening very closely on Thursday to American concerns about human tights…* and *… meet on Friday with his Chinese...*

The second type of problems cannot be solved at the macro level. These are individual problems that should be handled by writing specific rules for the related PV. An example is the possible spurious match of the PV *have…out* in the sentence *...still have our budget analysts out*

working the numbers. Since *have* is a verb with numerous usages, we should impose more individual constraints for NP insertion to prevent spurious matches, rather than calling a common macro shared by all Type II verbs.

4.4 Efficiency Testing

To test the efficiency of the index-based PV Expert Lexicon in comparison with a sequential Finite State Automaton (FSA) in the PV identification task, we conducted the following experiment.

The PV Expert Lexicon was compiled as a regular local grammar into a large automaton that contains 97,801 states and 237,302 transitions. For a file of 104 KB (the MUC-7 *dryrun* corpus of 16,878 words), our sequential FSA runner takes over 10 seconds for processing on the Windows NT platform with a Pentium PC. This processing only requires 0.36 second using the indexed PV Expert Lexicon module. This is about 30 times faster.

5 Conclusion

An effective and efficient approach to phrasal verb identification is presented. This approach handles both separable and inseparable phrasal verbs in English. An Expert Lexicon formalism is used to develop the entire phrasal verb lexicon and its associated pattern matching rules and macros. This formalism allows the phrasal verb lexicon to be called between two levels of parsing for the required morpho-syntactic interaction in phrasal verb identification. Benchmarking using both the running text corpus and sampling corpus shows that the presented approach provides a satisfactory solution to this problem.

In future research, we plan to extend the successful experiment on phrasal verbs to other types of multi-word expressions and idioms using the same expert lexicon formalism.

Acknowledgment

This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-03-C-0044. The authors wish to thank Carrie Pine and Sharon Walter of AFRL for supporting and reviewing this work. Thanks also go to the anonymous reviewers for their constructive comments.

References

Breidt. E., F. Segond and G. Valetto. 1994. Local Grammars for the Description of Multi-Word Lexemes and Their Automatic Recognition in Text. *Proceedings of Comlex-2380 - Papers in Computational Lexicography*, Linguistics Institute, HAS, Budapest, 19-28.

Breidt, *et al.* 1996. Formal description of Multi-word Lexemes with the Finite State formalism: IDAREX. *Proceedings of COLING 1996*, Copenhagen.

Bolinger, D. 1971. *The Phrasal Verb in English*. Cambridge, Mass., Harvard University Press.

Church, K. 1988. A stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of ANLP 1988*.

Di Sciullo, A.M. and E. Williams. 1987. *On The Definition of Word*. The MIT Press, Cambridge, Massachusetts.

Fraser, B. 1976. *The Verb Particle Combination in English*. New York: Academic Press.

Pelli, M. G. 1976. *Verb Particle Constructions in American English*. Zurich: Francke Verlag Bern.

Sag, I., T. Baldwin, F. Bond, A. Copestake and D. Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. *Proceedings of CICLING 2002*, Mexico City, Mexico, 1-15.

Shaked, N. 1994. *The Treatment of Phrasal Verbs in a Natural Language Processing System*, Dissertation, CUNY.

Silberztein, M. 2000. INTEX: An FST Toolbox. *Theoretical Computer Science*, Volume 231(1): 33-46.

Small, S. and C. Rieger. 1982. Parsing and comprehending with word experts (a theory and its realisation). W. Lehnert and M. Ringle, editors, *Strategies for Natural Language Processing*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Srihari, R., W. Li, C. Niu and T. Cornell. 2003. InfoXtract: An Information Discovery Engine Supported by New Levels of Information Extraction. *Proceeding of HLT-NAACL Workshop on Software Engineering and Architecture of Language Technology Systems*, Edmonton, Canada.

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