Identification of Verb-Particle Constructions in English

Ali Basirat
Department of Linguistics and Philology
Uppsala University
ali.basirat@lingfil.uu.se

Abstract

We propose different syntax-based methods for automatically identifying verb-particle constructions in English. The methods are based on the Deterministic Finite-state Automaton (DFA), Hidden Markov Model(HMM), and Synchronous Context-Free Grammar (SCFG). Our experiments show that the methods could result in F-score 83.3% over our manually annotated test-set consisting of Wikipedia articles and British National Corpus (BNC).

1 Introduction

Multiword expressions (MWE) are lexical items that are made up of multiple simplex words (e.g., nominal compounds, idioms, and verb-particle constructions) (Kim and Baldwin, 2010). These units can be characterized as "idiosyncratic interpretations that cross word boundaries" (Sag et al., 2002). The two main problems relating to MWEs are (1) identification, and (2) extraction (Baldwin and Kim, 2010). The problem of MWE identification deals with distinguishing between MWEs and literal words. The MWE extraction, however, is a lexicon development task which tries to extract MWE lexical items from a corpus.

In this paper, we study three models for identifying verb-particle constructions (VPC) in English, a particular MWE made up of a head verb and one or more obligatory particles (Baldwin, 2005). Particles in a VPC can appear in three forms, (1) intransitive preposition, (2) adjective, and (3) verb. The purpose of this research is to study different syntax-based methods for identifying prepositional VPCs in English texts.

The set of VPCs extracted by Baldwin, 2005) are used as the desired VPCs that our models are expected to identify. The proposed models are basically based on (1) a deterministic finite state automaton, (2) a hidden Markov model, and (3) a synchronous context-free grammar. Using these models, we could obtain the F1-score of 83.3% on our test set.

In the remainder of this paper, we give a short history of previous work has been done in identifying English VPCs. Sec. 3 outlines the data we used for training and testing our models. Then we detail the VPC identification models in Sec. 4, Sec. 5, and Sec. 6. In Sec. 7, we represents the results obtained from the combination of the output of the models.

2 Related Work

Baldwin and Villavicencio, 2002) and Baldwin, 2005) propose a series of unsupervised, weakly supervised, and supervised techniques for extracting and identifying VPCs from texts. Their methods were based on tagger output, chuncker output, and a chunker grammar and parser output. In the best case, their methods could result in an F-score 0.749 for intransitive and 0.879 for transitive verbs on a manually annotated test set collected from the British national Corpus (BNC). They uses BNC, WSJ, and the Brown corpus for training.

McCarthy et al., 2003) used the RASP parser to identify MWEs such as noun compounds or VPCs. The extracted VPCs were used in a MWE composi-

tionality study. Their model could result in 75.82% of F1-score on the Wall Street Journal (WSJ).

Kim and Baldwin, 2010) propose a method for automatically identifying VPCs in raw text. Using the RASP parser and WordNet 2.1, they extracted both syntactic and semantic features of the words and built a supervised classifier using TIMBL. The syntactic information retrieved from the RASP out were the verb-lemma, preposition, and the head noun of the subject and object of each verb. They also obtained lexical semantics of the head nouns based on the WordNet. They test three strategies for dealing with errors in detecting pronouns, proper nouns, and WH words as noun. The first strategy was to resolved all the errors manually. In this case their model could result in 97.4% of F1-score. In the second strategy, all errors were left unresolved and in the third strategy, the proper nouns were replaced with hypernyms. In these cases their model result in 95.9% of F1-score.

The difference between the test-sets used in the aforementioned research and our test-set makes direct comparison with this research difficult. McCarthy et al., 2003) evaluated their methods relative to the WSJ corpus. Baldwin, 2005) used a manually annotated test set which was collected from the BNC. However, as will be outlined in the next section, our test set is collected from Wikipedia and the VPC-annotated corpus released by Baldwin, 2005).

3 Resources

Our experiments were carried out on sentences we have collected from two corpora. The first corpus has been released by Baldwin, 2005) contains $23,600$ sentences classified into 506 classes corresponding to different English VPCs. We will refer to this corpus as *Baldwin-corpus*. Almost all of the sentences in the corpus contain at least one VPC (positive sample) to which the sentence belongs. We used a simple deterministic finite state automaton (DFA) to annotate the sentence with the VPCs.

All sentences in the Baldwin-corpus contain at least one VPC (positive sample), hence the need for the second corpus that contains sentences without any VPC (negative samples). The second corpus was collected from Wikipedia articles. The corpus contains $30,000$ sentences. We will refer to this corpus as *Wiki-corpus*.

The Wiki-corpus was annotated in three steps. First, it is tagged using the Stanford part-of-speech tagger (Toutanova et al., 2003). Second, each word pair (w_1, w_2) with maximum word-distance 5 words which is tagged as (VB,RP) and matches one of the verb-particles extracted by Baldwin, 2005) is selected as a candidate VPC. In the third step, the candidate VPCs are manually examined.

There is no doubt that this method cannot find all VPCs in the Wiki-corpus. Using this method, verb-particle constructions could be identified in only 0.6% of sentences in the corpus. The method is limited to the error involved in the part-of-speech tagger and the list of VPCs extracted by Baldwin, 2005). We can expect an acceptable value of recall for the extracted VPCs because they were manually examined in the third step of the annotation process. However, the precision can be very low. The precision of this method on a small part of Wiki-corpus containing around 300 sentences is 50%.

We accept the noisy data in the corpus and use it only in the training phase in our models because the only reason why we use this corpus is the sentences that have not included verb-particle constructions.

We have combined both Wiki-corpus and Baldwin-corpus and split the resulting corpus into three parts used in training, development, and testing VPC identification models. Table 1 shows the number of sentences used in train, development, and test sets.

Table 1: Number of sentences in the train, development, and test data

	Train	Development	Test
Wiki-corpus	23663	6000	337
Baldwin-corpus	18882	4500	218
All	42545	10500	555

In order to preserve the validity of results we have manually annotated the test set. Our manual annotations for the Baldwin-corpus part of the test data does not completely match the original annotation of the data. In the original annotation, only one VPC in each sentence in marked, but we could find some sentences that contain two VPCs. For the Wiki-corpus part of the test data we could find some VPCs that were not listed by Baldwin, 2005). Table

2 shows the distribution of VPCs over the corpora.

Table 2: Distribution of VPCs on the train, development, and test corpora

	Train	Development	Test
Wiki-corpus	0.007	0.001	0.05
Baldwin-corpus	0.93	0.93	0.98
All	0.41	0.41	0.42

4 DFA Model

We have examined three crude deterministic finite state automata (DFA) for identifying verb-particle constructions. The first model is a tag-based DFA in which VPCs are identified by looking at part-of-speech tags of words. This model closely follows the tag-based VPC extraction method introduced in Baldwin, 2005). The verb-particle constructions in the DFA model are identified only by looking at the part-of-speech tag of the word. Every particle that follows a verb with maximum word-distance 5 words is assumed as a verb-particle construction.

The second model is just like the first model but instead of looking at POS tags we look at word stems. In this model, every paired words w_1 and w_2 with maximum word distance 5 words is considered as a VPC if their stems are in the list of VPCs extracted by Baldwin, 2005).

The third model is a combination of the first model and second model. In this model every paired words w_1 and w_2 that satisfies the following conditions is considered as a VPC.

1. w_1 is tagged as *verb* and w_2 is tagged as *particle*

2. The combination of w_1 and w_2 must be in the set of VPCs extracted by Baldwin, 2005).

3. The word-distance between w_1 and w_2 must not exceed 5 words.

The results for DFA-based identification models are presented in Table 3. The table shows big difference between the recalls. This is because of the existence error in the output of the part-of-speech tagger.

Table 3: VPC identification results obtained from deterministic finite automata

	Precision	Recall	F-Score
Tag-based	0.58	0.26	0.36
Word-based	0.84	**0.70**	**0.76**
Combined	**0.90**	0.45	0.60

5 Hidden Markov Model

The DFA models discussed in Sec. 4 does not include any statistical information in the process of identifying verb-particle constructions. In this section, we examine how adding statistical information to the DFAs can affect the quality of the models.

A Hidden-Markov Model (HMM) (Rabiner, 1990) corresponds to an Stochastic Finite-state Automaton (SFA) in which state transitions and observations are based on some probability distribution functions. The HMM is denoted (N, M, A, B, P) where N is number of hidden stated; M is number of possible observations; A is transition probabilities; B is observation probabilities; P is initial state distribution.

For the problem of identifying VPCs, each word in a sentence can be in one of the following three states, which correspond to the hidden states of the HMM:

1. S_1: The word is not a part of a VPC

2. S_2: The word is head-word of a VPC

3. S_3: The word is inside a VPC

Different word representations (e.g., word stem, and part-of-speech tag) can be used as observation symbols of the HMM. Given the set of hidden states, we have trained two HMMs based on (1) word-stems, and (2) part-of-speech tags. The observation probabilities in the word-based model at time t are $Pr(O_t = s(w_i)|q_t = S_j)$, where s is the stemming function returning the stem of the lexical item w, and O_t and q_t are the random variables denoting the observation and the hidden state at time t respectively. Similarly, the observation probabilities in the tag-based model at time t are $Pr(O_t = p(w_i)|q_t = S_j)$, where p is the function returning the POS-tag of the lexical item w.

Given an input sentence, the process of VPC identification using the HMMs can be carried out in two steps:

1. Word-representation

2. HMM decoding

The word-representation step in the word-based model and the POS-based model corresponds to the stemming and part-of-speech tagging the input sentence. In the HMM decoding we apply the standard HMM decoding algorithm, called *Viterbi*, on the output of the word-representation step.

Table 4 represents the VPC identification results over the manually annotated test set. The high value of precision in the tag-based model shows that most of the VPCs identified by the model match the VPCs in the test set. The low value of recall, however, shows that there are many VPCs in the test set that could not be identified by the tag-based model. Our experiments show that around 30% of the errors in the tag-based model are because of the error involved in the output of the POS-tagger. 17% of errors has also been because of the limitation of the HMM in modeling discontinuous VPC.

The word-based model sacrifices precision for improving recall. The main sources of errors in the word-based models are (1) the discontinuous VPCs, (2) the out-of-vocabulary VPCs. Around 31% of false-negative errors in the word-based models are because of the natural limitation of the HMM that could not model the discontinuous VPCs. These errors have almost been occurred in the same sentences in both the word-based model and the tag-based model. The out-of-vocabulary VPCs are those that have not been seen in the training data but in the test data. Most of these VPCs are in the wiki-corpus part of the test data. As mentioned in Sec. 3 around half of the VPCs in the wiki-corpus are not in the list of VPCs marked in the Baldwin-corpus.

The rows *Intersection*, and *Union* are related to the combination of word-based and tag-based models. In the Intersection model, a VPC is a sequence of words that both word-based and tag-based models mark it as a VPC. In the Union model, a VPC is a sequence of words that at least one of tag-based or word-based models marks it as a VPC. As shown in Table 4, the best values of recall and F-score are for the Union model.

Table 4: VPC identification results obtained from the Hidden Markov Models

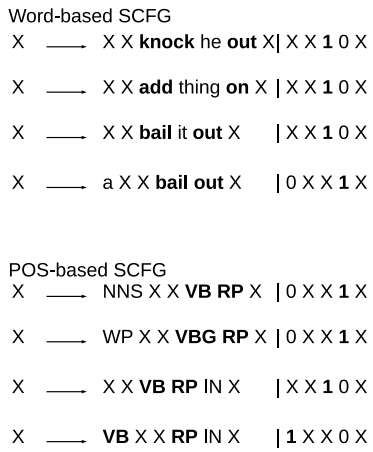
	Precision	Recall	F
Word-based	0.85	0.73	0.78
Tag-based	0.90	0.54	0.67
Intersection	**0.96**	0.48	0.64
Union	0.82	**0.78**	**0.80**

6 Synchronous Context-Free Grammar

In this section we examine a sequence classifier model that is able to handle both continuous and discontinuous VPCs. The model is based on the synchronous context-free grammar (SCFG), a generalization of the context-free grammar (CFG). An SCFG is denoted $G = (V, \Sigma, \Delta, R, S)$, where V is the set of variables (non-terminals), Σ and Δ are the set of words (alphabet symbols or terminals) in the source (input) and target (output) languages, respectively, R is a finite set of production rules, and S in V is the start symbol. Each rule r in R is an object $A \rightarrow (\alpha, \beta, \Pi)$, where A is a variable in V, α is in $(V \cup \Sigma)^*$, β is in $(V \cup \Delta)^*$, and Π is a permutation which corresponds the objects in α to the objects in β. Productions in a *statistical SCFG* are weighted with probabilities. There are many ways to add the statistical information to the rules. For instance, the weight of a production $A \rightarrow (\alpha, \beta, \Pi)$ can be: (1) the probability of co-occurrence of two parts of the right-hand side given the left hand side $P(\alpha, \beta \mid A)$, or (2) the probability of occurrence of target part of the right-hand side given the co-occurrence of the left-hand side and source part of the right-hand side $P(\beta \mid A, \alpha)$.

The problem of identifying verb-particle construction can be seen as a translation problem in which each sequence of words is translated into one of two symbols 0, which means the sequence is not a VPC, or 1, which means the sequences is a VPC. The alphabet symbols in the source language Σ in this formulation can be any representation of words (e.g., raw words, word stems, POS-tag, supertag). The target language, however, has only two symbols $\{0, 1\}$. The grammar has only two non-terminals S,

Figure 1: The word-based and POS-based synchronous context-free grammars used for VPC identification



which is used an start symbol, and X, which is used as a place holder.

Given such an SCFG, the problem of identifying VPCs can be carried out in two steps:

1. Representation of words in the source language alphabet
2. Parsing in the SCFG

We have extracted two statistical SCFG from the training data, a word-based grammar and a POS-based grammar, using the grammar extraction approach proposed by Chiang, 2007) and implemented in Moses (Koehn et al., 2007). The set of terminal symbols Σ in the word-based grammar contains the English word stems, and in the POS-based grammar, it contains the Penn POS tags. Δ in both grammars is the same. Fig. 1 shows some sample rules in the extracted SCFGs. The bold items in the source and target sides of the rules are aligned to each other. As shown, the grammars could extract some rules for both continuous and discontinuous VPC.

Table 5 represents the evaluation results obtained from the grammar-based VPC identifiers. Rows *Intersection* and *Union* have the same interpretation as in Sec. 5.

7 Combined Model

Table 6 represents the evaluation results obtained from the intersection and union of the best outputs achieved from the VPC identification models. As shown, best F-scores are obtained from DFA $\cup$ HMM, and with a slight change from DFA $\cup$ HMM $\cup$ SCFG. Our experiments show that around 34% and 66% of errors have been occurred in the Wiki-corpus and the Baldwin-corpus part of the test set respectively. It means that the models could not correctly identify VPCs in 65% of the Wiki-corpus sentences and 1% of the Baldwin-corpus sentences. It shows that most of the errors are related to the VPCs which were no annotated in the train data but used in the test data.

Table 5: VPC identification results obtained from the word-based and POS-based SCFGs

	Precision	Recall	F
Word-based	0.89	0.66	0.75
Tag-based	0.9	0.57	0.69
Intersection	**0.93**	0.47	0.62
Union	0.87	**0.76**	**0.81**

Table 6: Evaluation results optained from the combination of outputs of the DFA, HMM, and SCFG

		Precision	Recall	F-Score
Union	DFA $\cup$ HMM	0.80	0.87	**0.833**
	DFA $\cup$ SCFG	0.83	0.82	0.825
	HMM $\cup$ SCFG	0.82	0.83	0.825
	DFA $\cup$ HMM $\cup$ SCFG	0.79	**0.88**	**0.832**
Intersection	DFA $\cap$ HMM	0.89	0.60	0.716
	DFA $\cap$ SCFG	**0.90**	0.65	0.754
	HMM $\cap$ SCFG	0.88	0.71	0.785
	DFA $\cap$ HMM $\cap$ SCFG	0.88	0.71	0.785

8 Conclusion

This paper has focused on the problem of identifying verb-particle constructions (VPCs) in English. Different models were proposed based on (1) deterministic finite state automaton, (2) hidden Markov model, and (3) synchronous context-free grammar. All models were examined based on word-stems and part-of-speech tags of the words. The combination of the best outputs of the models results in the F1-score of 83.3% over our manually annotated test set collected from the Wikipedia articles and BNC.

In future research, we interested in modeling different classes of MWE using syntactic features of

the words. We believe that the relationship between the different components of MWE can be efficiently modeled using a synchronous context-free-grammar. In addition, it seems that the synchronous context-free grammar used in this research can be replaced by a simple context-free grammar in which the non-terminal are the symbols corresponding to the different classes of MWE and the terminals are basic representations of words (e.g., POS-tag, word-stem, or supertag).

References

Baldwin, T. (2005). Deep lexical acquisition of verb–particle constructions. *Computer Speech & Language*, 19(4):398–414.

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. *Handbook of Natural Language Processing, second edition. Morgan and Claypool*.

Baldwin, T. and Villavicencio, A. (2002). Extracting the unextractable: A case study on verb-particles. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

Kim, S. N. and Baldwin, T. (2010). How to pick out token instances of english verb-particle constructions. *Language resources and evaluation*, 44(1-2):97–113.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

McCarthy, D., Keller, B., and Carroll, J. (2003). Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*, pages 73–80. Association for Computational Linguistics.

Rabiner, L. R. (1990). Readings in speech recognition. chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15. Springer.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.