Learning Verb Argument Structure from Minimally Annotated Corpora*

Anoop Sarkar and Woottiporn Tripasai
Dept. of Computer and Information Science
University of Pennsylvania
200 South 33rd Street,
Philadelphia, PA 19104-6389 USA
{anoop,tripasai}@linc.cis.upenn.edu

Abstract

In this paper we investigate the task of automatically identifying the correct argument structure for a set of verbs. The argument structure of a verb allows us to predict the relationship between the syntactic arguments of a verb and their role in the underlying lexical semantics of the verb. Following the method described in (Merlo and Stevenson, 2001), we exploit the distributions of some selected features from the local context of a verb. These features were extracted from a 23M word WSJ corpus based on part-of-speech tags and phrasal chunks alone. We constructed several decision tree classifiers trained on this data. The best performing classifier achieved an error rate of 33.4%. This work shows that a subcategorization frame (SF) learning algorithm previously applied to Czech (Sarkar and Zeman, 2000) is used to extract SFs in English. The extracted SFs are evaluated by classifying verbs into verb alternation classes.

1 Introduction

The classification of verbs based on their underlying thematic structure involves distinguishing verbs that take the same number and category of arguments but assign different thematic roles to these arguments. This is often termed as the classification of verb diathesis roles or the lexical semantics of predicates in natural language (see (Levin, 1993; McCarthy and Korhonen, 1998; Stevenson and Merlo, 1999; Stevenson et al., 1999; Lapata, 1999; Lapata and Brew, 1999; Schulte im Walde, 2000)). Following the method described in (Merlo and Stevenson, 2001; Stevenson and Merlo, 1999; Stevenson et al., 1999), we exploit the distributions of some selected features from the local context of a verb but we differ from these previous studies in the use of minimally annotated data to construct our classifier. The data we use is only passed through a part-of-speech tagger and a chunker which is used to identify base phrasal categories such as noun-phrase and verb-phrase chunks to identify potential arguments of each verb.

Lexical knowledge acquisition plays an important role in corpus-based NLP. Knowledge of verb selectional preferences and verb subcategorization frames (SFs) can be extracted from corpora for use in various NLP tasks. However, knowledge of SFs is often not fine-grained enough to distinguish various verbs and the kinds of arguments that they can select. We consider a difficult task in lexical knowledge acquisition: that of finding the underlying argument structure which can be used to relate the observed list of SFs of a particular verb. The task involves identifying the roles assigned by the verb to its arguments. Consider the following verbs, each occuring with intransitive and transitive SFs[1].

Unergative

(1) a. The horse raced past the barn.
b. The jockey raced the horse past the barn.

Unaccusative

(2) a. The butter melted in the pan.
b. The cook melted the butter in the pan.

* This research was supported in part by NSF grant SBR-89-20230. Thanks to Paola Merlo, Dan Gildea, David Chiang, Aravind Joshi and the anonymous reviewers for their comments. Also thanks to Virginie Nanta for an earlier collaboration with the first author on an unsupervised version of this work.

[1] The examples are taken from (Merlo and Stevenson, 2001). See (Levin, 1993) for more information. The particular categorization that we use here is motivated in (Stevenson and Merlo, 1997)

Object-Drop

(3) a. The boy washed.
b. The boy washed the hall.

Each of the verbs above occurs with both the intransitive and transitive SFs. However, the verbs differ in their underlying argument structure. Each verb assigns a different role to their arguments in the two subcategorization possibilities. For each verb above, the following lists the roles assigned to each of the noun phrase arguments in the SFs permitted for the verb. This information can be used for extracting appropriate information about the relationships between the verb and its arguments.

Unergative

INTRAN: NP_{agent} raced

TRAN: NP_{causer} raced NP_{agent}

Unaccusative

INTRAN: NP_{theme} melted

TRAN: NP_{causer} melted NP_{theme}

Object-Drop

INTRAN: NP_{agent} washed

TRAN: NP_{agent} washed NP_{theme}

Our task is to identify the transitive and intransitive usage of a particular verb as being related via this notion of argument structure. This is called the *argument structure classification* of the verb. In the remainder of this paper we will look at the problem of placing verbs into such classes automatically.

Our results in this paper serve as a replication and extension of the results in (Merlo and Stevenson, 2001). Our main contribution in this paper is to show that a subcategorization frame (SF) learning algorithm previously applied to Czech (Sarkar and Zeman, 2000) can be applied to English and evaluated by classifying verbs into verb alternation classes. We perform this task using only tagged and chunked data as input to our subcategorization frame learning stage. Our result can be compared to previous work (Merlo and Stevenson, 2001) which did not use SF learning but used a 65M word WSJ corpus which was tagged as well as automatically parsed with a Treebank trained statistical parser. It is important to note that (Merlo and Stevenson, 2001) extract some features using the tagged information (in fact, those features that we use SF learning to extract) and other features using parse trees.

2 The Hypothesis

We create a probabilistic classifier that can automatically classify a set of verbs into argument structure classes with a reasonable error rate. We use the hypothesis introduced by (Stevenson and Merlo, 1999) that although a verb in a particular class can occur in all of the syntactic contexts as verbs from other classes the statistical distributions can be distinguished. In other words, verbs from certain classes will be more likely to occur in some syntactic contexts than others. We identify features that pick out the verb occurences in these contexts. By using these features, we will attempt to determine the classification of those verbs. In the previous section we saw that we sometimes have noun-phrase arguments (NP_{causer}) as being a *causer* of the action denoted by the verb. For example, (Stevenson and Merlo, 1999) show that a classifier can exploit these causativity facts to improve classifiction.

We use some new features in addition to the ones proposed and used in (Merlo and Stevenson, 2001) for this task. In addition, we include as a feature the probabilistic classification of the verb as a transitive or intransitive verb. Thus the classifier is simulaneously placing each verb into the appropriate subcategorization frame as well as identifying the underlying thematic roles of the verb arguments.

In our experiment, we will consider the following set of classes (each of these were explained in the previous section): unergative, unaccusative, and object-drop. We test 76 verbs taken from (Levin, 1993) that are in one of these three classes. The particular verbs were chosen to include high frequency as well as low frequency verb tokens in our particular corpus of 23M words of WSJ text.[2] We used all instances of these verbs from the WSJ corpus. The data was annotated with the right classification for each verb and the classifier was trained on 90% of the verbs taken from the 23M word corpus and tested on 10% of the data using 10-fold cross validation. We describe the experiment in greater detail

[2] The particular verbs selected were looked up in (Levin, 1993) and the class for each verb in the classification system defined in (Stevenson and Merlo, 1997) was selected with some discussion with linguists.

in Section 4.

3 Identifying subcategorization frames

An important part of identifying the argument structure of the verb is to find the verb's subcategorization frame (SF). For this paper, we are interested in whether the verb takes an intransitive SF or a transitive SF.

In general, the problem of identifying subcategorization frames is to distinguish between arguments and adjuncts among the constituents modifying a verb. For example, in "John saw Mary yesterday at the station", only "John" and "Mary" are required arguments while the other constituents are optional (adjuncts).[3]

The problem of SF identification using statistical methods has had a rich discussion in the literature (Ushioda et al., 1993; Manning, 1993; Briscoe and Carroll, 1997; Brent, 1994) (also see the refences cited in (Sarkar and Zeman, 2000)). In this paper, we use the method of hypothesis testing to discover the SF for a given verb (Brent, 1994). Along with the techniques given in these papers, (Sarkar and Zeman, 2000; Korhonen et al., 2000) also discuss other methods for hypothesis testing such the use of the t-score statistic and the likelihood ratio test. After experimenting with all three of these methods we selected the likelihood ratio test because it performed with higher accuracy on a small set of hand-annotated instances. We use the determination of the verb's SF as an input to our argument structure classifier (see Section 4).

The method works as follows: for each verb, we need to associate a score to the hypothesis that a particular set of dependents of the verb are arguments of that verb. In other words, we need to assign a value to the hypothesis that the observed frame under consideration is the verb's SF. Intuitively, we either want to test for independence of the observed frame and verb distributions in the data, or we want to test how likely is a frame to be observed with a particular verb without being a valid SF. We develop these intuitions by using the method of hypothesis testing using the likelihood ratio test. For further background on this method of hypothesis testing the reader is referred to (Bickel and Doksum, 1977; Dunning, 1993).

3.1 Likelihood ratio test

Let us take the hypothesis that the distribution of an observed frame f in the training data is independent of the distribution of a verb v. We can phrase this hypothesis as $p(f \mid v) = p(f \mid !v) = p(f)$, that is distribution of a frame f given that a verb v is present is the same as the distribution of f given that v is not present (written as $!v$). We use the log likelihood test statistic (Bickel and Doksum, 1977, 209) as a measure to discover particular frames and verbs that are highly associated in the training data.

$$\begin{aligned} k_1 &= c(f, v) \\ n_1 &= c(v) = c(f, v) + c(!f, v) \\ k_2 &= c(f, !v) \\ n_2 &= c(!v) = c(f, !v) + c(!f, !v) \end{aligned}$$

where $c(\cdot)$ are counts in the training data. Using the values computed above:

$$\begin{aligned} p_1 &= \frac{k_1}{n_1} \\ p_2 &= \frac{k_2}{n_2} \\ p &= \frac{k_1 + k_2}{n_1 + n_2} \end{aligned}$$

Taking these probabilities to be binomially distributed, the log likelihood statistic (Dunning, 1993) is given by:

$$\begin{aligned} &-2 \log \lambda = \\ &\quad 2[\log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) - \\ &\quad \log L(p, k_1, n_2) - \log L(p, k_2, n_2)] \end{aligned}$$

where,

$$\log L(p, n, k) = k \log p + (n - k) \log(1 - p)$$

According to this statistic, the greater the value of $-2 \log \lambda$ for a particular pair of observed frame and verb, the more likely that frame is to be valid SF of the verb. If this value is above a certain threshold it is taken to be a positive value for the binary feature

[3] There is some controversy as to the *correct* subcategorization of a given verb and linguists often disagree as to what is the right set of SFs for a given verb. A machine learning approach such as the one followed in this paper sidesteps this issue altogether, since it is left to the algorithm to learn what is an appropriate SF for a verb. The stance taken in this paper is that the efficacy of SF learning is evaluated on some domain, as is done here on learning verb alternations.

TRAN, else it is a positive feature for the binary feature INTRAN in the construction of the classifier.[4]

4 Steps in Constructing the Classifier

To construct the classifier, we will identify features that can be used to accurately distinguish verbs into different classes. The features are computed to be the probability of observing a particular feature with each verb to be classified. We use C5.0 (Quinlan, 1992) to generate the decision tree classifier. The features are extracted from a 23M word corpus of WSJ text (LDC WSJ 1988 collection). Note that the training and test data constructed from this set are produced by the classification of individual verbs into their respective classes taken from (Merlo and Stevenson, 2001).

We prepare the corpus by passing it through Adwait Ratnaparkhi's part-of-speech tagger (Ratnaparkhi, 1996) (trained on the Penn Treebank WSJ corpus) and then running Steve Abney's chunker (Abney, 1997) over the entire text. The output of this stage and the input to our feature extractor is shown below.

```
Pierre         NNP   nx   2
Vinken         NNP
,              ,
61             CD    ax   3
years          NNS
old            JJ
,              ,
will           MD    vx   2
join           VB
the            DT    nx   2
board          NN
as             IN
a              DT    nx   3
nonexecutive   JJ
director       NN
Nov.           NNP
29             CD
.              .
```

We use the following features to construct the classifier. The first four features were discussed and motivated in (Stevenson and Merlo, 1999; Merlo and Stevenson, 2001). In some cases, we have modified the features to include information about part-of-speech tags. The discussion below clarifies the similarities and changes. The features we used in addition are the last two in the following list, the part-of-speech features and the subcategorization frame features. [5]

1. simple past (VBD), and past participle(VBN)
2. active (ACT) and passive (PASS)
3. causative (CAUS)
4. animacy (ANIM)
5. Part of Speech of the subject noun-phrase and object noun-phrase
6. transitive (TRAN) and intransitive (INTRAN)

To calculate all the probability values of each features, we perform the following steps.

4.1 Finding the main verb of the sentences

To find the main verb, we constructed a deterministic finite-state automaton that finds the main verb within the verb phrase chunks. This DFA is used in two steps. First, to select a set of main verbs from which we select the final set of 76 verbs used in our experiment. Secondly, the actual set of verbs is incorporated into the DFA in the feature selection step.

4.2 Obtaining the frequency distribution of the features

The general form of the equation we use to find the frequency distribution of each feature of the verb is the following:

$$P(V_j) = \frac{C(V_j)}{\sum_{1 \leq x \leq N} C(V_x)}$$

where $P(V_j)$ is the distribution of feature j of the verb, N is the total number of features of the particular type (e.g., the total number of CAUS features or ANIM features as described below) and $C(V_j)$ is the number of times this feature of the verb was observed in the corpus. The features computed using this formula are: ACT, PASS, TRAN, INTRAN, VBD, and VBN.

[4] See (Sarkar and Zeman, 2000) for information on how the threshold is selected.

[5] Note that while (Stevenson and Merlo, 1999; Merlo and Stevenson, 2001) used a TRAN/INTRAN feature, in their case it was estimated in a completely different way using tagged data. Hence, while we use the same name for the feature here, it is not the same kind of feature as the one used in the cited work.

4.3 The causative feature: CAUS

To correctly obtain the causative values of the testing verbs, we needed to know the meaning of the sentences. In this paper, we approximate the value by using the following approach. Also, the causative value is not a probability but a weight which is subsequently normalized.

We extract the subjects and objects of verbs and put them into two sets. We use the last noun of the subject noun phrase and object noun phrase (tagged by NN, NNS, NNP, or NNPS), as the subject and object of the sentences. Then the causative value is

$$\text{CAUS} = \frac{\text{overlap}}{\text{sum of all subject and objects in multiset}}$$

where the overlap is defined as the largest multiset of elements belonging to both subjects and objects multisets.

If subject is in the set $\{a, a, b, c\}$ and object is in set $\{a, d\}$, the intersection between both set will be $\{a, a\}$, and the causative value will be $\frac{2}{(4+2)} = \frac{1}{3}$.

If subject is in the set $\{a, a, b, c\}$ and object is in the set $\{a, b, d\}$, the intersection between both set will be $\{a, a, b\}$, and the causative value will be $\frac{(2+1)}{(4+3)} = \frac{3}{7}$.

Note that using this measure, we expect to get higher weights for tokens that occur frequently in the object position and sometimes in the subject position. For example, $\text{CAUS}(\{a, b\}, \{a, b\}) = \frac{2}{4}$ while $\text{CAUS}(\{a, b\}, \{a, a, a\}) = \frac{3}{5}$. This difference in the weight given by the CAUS feature is exploited in the classifier.

4.4 The animate feature: ANIM

Similar to CAUS, we can only approximate the value of animacy. We use the following formula to find the value:

ANIM = number of occurrence of pronoun in subject/number of occurrence of verbs

The set of pronouns used are *I, we, you, she, he,* and *they*. In addition we use the set of part-of-speech tags which are associated with animacy in Penn Treebank tagset as part of set of features described in the next section.

4.5 Part of Speech of object and subject

The part-of-speech feature picks up several subtle cues about the differences in the types of arguments selected by the verb in its subject or object position.

We count the occurrence of the head nouns of the subject noun phrase and the object noun phrase. Then, we find the frequency distribution by using the same formula as before:

$$P(V_j) = \frac{C(V_j)}{\sum_{1 \leq x \leq N} C(V_x)}$$

where $P(V_j)$ is the distribution of part of speech j, N is the total number of relevant POS features and $C(V_j)$ is the number of occurrences of part of speech j. Also, we limit the part of speech to only the following tags of speech: NNP, NNPS, EX, PRP, and SUCH, where NNP is singular noun phrase, NNPS is plural noun phrase, EX is ‘there’, PRP is personal pronoun, and SUCH is ‘such’.

4.6 Transitive and intransitive SF of the verb

To find values for this feature we use the technique described in Section 3. For each verb in our list we extract all the subsequent NP and PP chunks and their heads from the chunker output. We then perform subcategorization frame learning with all subsets of these extracted potential arguments. The counts are appropriately assigned to these subsets to provide a well-defined model. Using these counts and the methods in Section 3 we categorize a verb as either transitive or intransitive. For simplicity, any number of arguments above zero is considered to be a candidate for transitivity.

4.7 Constructing the Classifier

After we obtain all the probabilistic distributions of the features of our testing verbs, we then use C5.0 (Quinlan, 1992) to construct the classifier. The data was annotated with the right classification for each verb and the classifier was run on 10% of the data using 10-fold cross validation.

5 Results

We tried all possible feature combinations (individual features and all possible conjunctions of those features) to explore the contributions of each feature to the reduction of the error rate. The following are the results of the best performing feature combinations.

With our base features, ACT, PASS, VBD, VBN, TRAN, and INTRAN we get the average error rate of 49.4% for 10 fold cross validation. We can see that when we add the CAUS feature, the average error decreases to 41.1%. The CAUS feature helps in decreasing the error rate. Also, when we add the ANIM feature, we get a much better performance. Our average error rate decreases to 37.5%.

Features	Average error rate from Decision Tree	SE	Average error rate from Rule Set	SE
TRAN, INTRAN, VBD, VBN, PASS, ACT	49.4%	1.1%	67.7%	0.9%
TRAN, INTRAN, VBD, VBN, PASS, ACT, CAUS	41.1%	0.8%	40.8%	0.6%
TRAN, INTRAN, VBD, VBN, PASS, ACT, ANIM	37.5%	0.8%	36.9%	1.0%
TRAN, INTRAN, VBD, VBN, PASS, ACT, PART OF SPEECH	39.2%	0.8%	38.1%	1.1%
TRAN, INTRAN, VBD, VBN, PASS, ACT, CAUS, ANIM	**33.4%**	**0.7%**	**33.9%**	**0.8%**
TRAN, INTRAN, VBD, VBN, PASS, ACT, CAUS, PART OF SPEECH	39.0%	0.7%	37.1%	0.9%
TRAN, INTRAN, VBD, VBN, PASS, ACT, ANIM, PART OF SPEECH	35.8%	1.3%	35.9%	1.7%
TRAN, INTRAN, VBD, VBN, PASS, ACT, CAUS, ANIM, PART OF SPEECH	39.5%	1.0%	38.3%	1.0%

Figure 1: Results of the verb classification. Bold face results are for the best performing set of features in the classifier.

This is the lowest error rate we can achieve by adding one extra feature in addition to the base features. The ANIM feature is an important feature that we can use to construct the classifier. When we add the PART OF SPEECH feature, the error rate also decreases to 39.2%. Therefore, the PART OF SPEECH also helps reduce the error rate as well. When we put together the CAUS feature and ANIM feature, we achieve the lowest error rate, which is 33.4%. When we put the PART OF SPEECH and CAUS features together, the error rate does not really decrease (39.0%), comparing to the result with only the PART OF SPEECH feature. The reason of this result should be that there are some parts of the PART OF SPEECH feature and CAUS feature that overlap. When we add the ANIM and PART OF SPEECH features together, the error rate does decrease to 35.8%. Although the result is not as good as result of using ANIM and CAUS features, the combination of the ANIM and PART OF SPEECH features could be considered effective features that we can use to construct the classifier. We then combine all the features together. The result as expected is not very good. The error rate is 39.5%. The reason should be the same reason as the lower performance when combining the CAUS and PART OF SPEECH features.

Note that the features TRAN/INTRAN are needed for computing a large subset of the features used. Hence we did not conduct any experiments without these features. These experiments show that the use of SF learning can be useful to the performance of the verb alternation classifier. The error rate of the baseline classifier (picking the right argument structure at chance) was 65.5%. (Merlo and Stevenson, 2001) calculate the expert-based upper bound at this task to be an error rate of 13.5%.

Our best performing classifier achieves a 33.4% error rate. In comparison, (Merlo and Stevenson, 2001) obtain an error rate of 30.2% using a tagged and automatically parsed data set of 65M words of WSJ text. Thus, while we obtain a slightly worse error rate, this is obtained using a much smaller set of training data.

6 Conclusion

In this paper, we discussed a technique which automatically identified the correct argument structure of a set of verbs. Our results in this paper serve as a replication and extension of the results in (Merlo and Stevenson, 2001). Our main contribution in this paper is to show that with reasonable accuracy, this task can be accomplished using only tagged and chunked data. In addition, we incorporate some additional features such as part-of-speech tags and the use of subcategorization frame learning as part of our classification algorithm.

We exploited the distributions of selected features from the local context of the verb which was extracted from a 23M word WSJ corpus. We used C5.0 to construct a decision tree classifier using the values of those features. We were able to construct a classifier that has an error rate of 33.4%. This work shows that a subcategorization frame learning algorithm (Sarkar and Zeman, 2000) can be applied to the task of classifying verbs into verb alternation classes.

In future work, we would like to classify verbs into alternation classes on a per-token basis (as is done in the approach taken by Gildea (2002)) rather than the per-type we currently employ and also incorporate information about word senses in order to feasibly include verb alternation information in a statistical parser.

References

Steve Abney. 1997. Part of speech tagging and partial parsing. In S. Young and G. Bloothooft, editors, *Corpus based methods in language and speech*, pages 118–136. Dordrecht: Kluwer.

Peter Bickel and Kjell Doksum. 1977. *Mathematical Statistics*. Holden-Day Inc.

Michael Brent. 1994. Acquisition of subcategorization frames using aggregated evidence from local syntactic cues. *Lingua*, 92:433–470. Reprinted in Acquisition of the Lexicon, L. Gleitman and B. Landau (Eds.). MIT Press, Cambridge, MA.

Ted Briscoe and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th ANLP Conference*, pages 356–363, Washington, D.C. ACL.

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

Daniel Gildea. 2002. Probabilistic models of verb-argument structure. In *Proc. of COLING-2002*.

A. Korhonen, G. Gorrell, and D. McCarthy. 2000. Statistical filtering and subcategorization frame acquisition. In *Proceedings of EMNLP 2000*.

Maria Lapata and Chris Brew. 1999. Using subcategorization to resolve verb class ambiguity. In Pascale Fung and Joe Zhou, editors, *Proceedings of WVLC/EMNLP*, pages 266–274, 21-22 June.

Maria Lapata. 1999. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of 37th Meeting of ACL*, pages 397–404.

Beth Levin. 1993. *English Verb Classes and Alternations*. Chicago University Press, Chicago, IL.

Christopher D. Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Meeting of the ACL*, pages 235–242, Columbus, Ohio.

Diana McCarthy and Anna Korhonen. 1998. Detecting verbal participation in diathesis alternations. In *Proceedings of COLING/ACL-1998. Student Session*, pages 1493–1495.

Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distribution of argument structure. *Computational Linguistics*, 27(3):373–408.

J. Ross Quinlan. 1992. *C4.5: Programs for Machine Learning*. Series in Machine Learning. Morgan Kaufmann, San Mateo, CA.

A. Ratnaparkhi. 1996. A Maximum Entropy Part-Of-Speech Tagger. In *Proc. of the Empirical Methods in Natural Language Processing Conference*, University of Pennsylvania.

Anoop Sarkar and Daniel Zeman. 2000. Automatic extraction of subcategorization frames for czech. In *Proceedings of COLING-2000*.

Sabine Schulte im Walde. 2000. Clustering verbs semantically according to their alternation behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, Saarbrcken, Germany, August.

Suzanne Stevenson and Paola Merlo. 1997. Lexical structure and parsing complexity. *Language and Cognitive Processes*, 12(2).

Suzanne Stevenson and Paola Merlo. 1999. Automatic verb classification using distributions of grammatical features. In *Proceedings of EACL '99*, pages 45–52, Bergen, Norway, 8–12 June.

Suzanne Stevenson, Paola Merlo, Natalia Kariaeva, and Kamin Whitehouse. 1999. Supervised learning of lexical semantic verb classes using frequency distributions. In *Proceedings of SIGLEX99: Standardizing Lexical Resources*, College Park, Maryland.

Akira Ushioda, David A. Evans, Ted Gibson, and Alex Waibel. 1993. The automatic acquisition of frequencies of verb subcategorization frames from tagged corpora. In *Proc. of the Workshop on Acquisition of Lexical Knowledge from Text*, Columbus, OH.