Verb Classification – Machine Learning Experiments in Classifying Verbs into Semantic Classes

Bart Decadt and Walter Daelemans

Center for Dutch Language and Speech
Language Technology Group
University of Antwerp – Campus Drie Eiken
2610 Wilrijk – Belgium
{bart.decadt,walter.daelemans}@ua.ac.be

Abstract

This paper presents the results of our machine learning experiments in verb classification. Using Beth Levin's semantic classification of the English verbs as a gold standard, we (i) test the hypothesis that the syntactic behavior of a verb can be used to predict its semantic class, and (ii) investigate whether a robust shallow parser can provide the necessary syntactic information. With 277 verbs belonging to six of Levin's classes, we do *type classification* experiments using RIPPER, an inductive rule learner. Having only a set of n most likely subjects or objects as features, this machine learning algorithm is able to predict the correct class with $\pm$ 58% accuracy. This result is comparable with results from other researchers, like Merlo and Stevenson, Stevenson and Joanis, and Schulte im Walde.

1. Introduction

1.1. Overview

In this paper, we present the results of our machine learning experiments in verb classification. We will start by sketching the background of this line of research, starting with Beth Levin's manual classification of the English verbs (Levin, 1993), and linking our work to it. Next, we will show the gold standard – six classes from Beth Levin's verb classification – used for evaluating the outcome of our experiments. We continue by explaining how we gathered data from the British National Corpus (BNC), and how we presented the data to the machine learning algorithm (RIPPER) used in the experiments. Finally, we report and analyze the results, compare them with related work and present the lines of research we will follow in the near future.

1.2. Background

In 1993, Beth Levin published her Ph.D. thesis (Levin, 1993), in which she described her handcrafted semantic classification of the English verbs. Her – very simplified – hypothesis is that the semantics of a verb determine to a large extent its syntactic behavior. By analyzing the English verbs along some syntactic criteria – among others the sub-categorization frames in which the verbs appear – she manages to distinguish 49 semantically coherent classes.

Levin's work was a source of inspiration, and a possibility for evaluation, for computational linguists working on semantic (verb) classification. The main goals of this line of research are (i) trying to classify or cluster words – in this case verbs – automatically according to their semantics, and (ii) determining which features are informative for this task.

Research on verb classification will enable us to do *lexical acquisition* for verbs: it will help in making or extending a lexicon with, for example, information on the semantic class of a verb. Another possible benefit of verb classification is that these techniques will help us to decide on the sub-categorization frame, or other syntactic or semantic information, of *unknown* or *new* verbs.

1.3. Our Research

In our research, we aim at *type classification* of English verbs into Levin's classes. With type classification, we mean that we collect information for the verbs, and for each particular verb we merge this information into one data vector. Then, on the basis of the collected information, we try to predict the semantic class of an unknown or new verb.

The information we use to classify the verbs is provided by a *shallow parser*: in the experiments reported here, we limited the information to the subjects and objects of the verbs. This information is fairly easy to extract from a shallow parsed corpus: we did not need to develop (complex) heuristics.

2. The Gold Standard

From Levin's classification, we selected a subset of 6 classes, some of which are divided in subclasses. These classes contain 318 verbs, of which we used only 277, because for the remaining 41 verbs we did not find or found not enough data in our corpus. Some of these verbs are ambiguous and appear in two of the six classes. For practical reasons, we ignored this ambiguity in our experiments: we assume that the 'main' class of a verb is the first class it appears in.

The selected classes and subclasses, with the number given by Levin to that class between brackets, are:

- *verbs of contact by impact* (18), containing four subclasses:
  - *hit verbs* (18.1)
  - *swat verbs* (18.2)
  - *spank verbs* (18.3)
  - *non-agentive verbs of contact by impact* (18.4)
- *poke verbs* (19)
- *verbs of contact* (20)

- *verbs of cutting* (21), containing two subclasses:
  - *cut verbs* (21.1)
  - *carve verbs* (21.2)
- *verbs of combining and attaching* (22), containing five subclasses:
  - *mix verbs* (22.1)
  - *amalgamate verbs* (22.2)
  - *shake verbs* (22.3)
  - *tape verbs* (22.4)
  - *cling verbs* (22.5)
- *verbs of separating and disassembling* (23), containing four subclasses:
  - *separate verbs* (23.1)
  - *split verbs* (23.2)
  - *disassemble verbs* (23.3)
  - *differ verbs* (23.4)

Table 1 shows the distribution of the 318 verbs over the 6 classes and 17 subclasses, expressed in numbers and percentages, and also lists some example verbs. Some classes are very small, like class 19, 22.5 and 23.4: machine learning algorithms can be expected to have difficulties learning these classes.

We evaluated our machine learning experiments in two ways. A first evaluation was done by looking at only the main classes: we will call this the *coarse-grained evaluation*. A second evaluation was done by taking the subclasses into account: we will call this the *fine-grained evaluation*.

From Table 1, we can induce a *random* and *default* baseline to compare our results with. The random baseline result is obtained by assigning class labels to the verbs according to the distribution in Table 1: in the coarse-grained case this results in 29.8% accuracy and in the fine-grained case in 9.4% accuracy. For the default baseline, we label each verb with the most frequent class label: this is class 22 in the coarse-grained case, resulting in 43.3% accuracy, and class 22.4 in the fine-grained case, resulting in 16.2% accuracy.

3. Data Acquisition and Representation

For the 277 verbs in the six classes from Levin, we collected information in the written part of the BNC ($\pm$ 60M words). This corpus was shallow parsed with a memory-based shallow parser (Buchholz et al., 1999; Daelemans et al., 1999), developed at our research site[1]. After shallow parsing, we were able to make two lists for each verb: one with all the head nouns of the subjects and one with all the head nouns of the objects. These two lists were sorted by the statistical measure *likelihood ratio*: with this measure, the following two hypotheses for a subject-verb or object-verb pair are examined – see also (Manning and Schütze, 1999):

- Hypothesis 1 is the formulation of independence: the fact that the noun occurs in the subject position is not heavily determined by the verb.

$$H_1 : P(\text{noun as subject}|\text{verb}) = p = P(\text{noun as subject}|\neg\text{verb})$$

- Hypothesis 2 is the formulation of dependence: the fact that the noun occurs in the subject position is to a large extent determined by the verb.

$$H_2 : P(\text{noun as subject}|\text{verb}) = p_1 \neq p_2 = P(\text{noun as subject}|\neg\text{verb})$$

The values for p, p_1 and p_2 are computed as follows:

$$s = f(\text{noun as subject})$$

$$sv = f(\text{noun as subject,verb})$$

$$v = f(\text{verb}) \qquad V = f(\text{all verbs})$$

$$p = \frac{s}{V} \qquad p_1 = \frac{sv}{v} \qquad p_2 = \frac{s - sv}{V - v}$$

Assuming a binomial distribution:

$$b(k; n, x) = \binom{n}{k} x^k (1 - x)^{(n-k)} \tag{1}$$

the likelihoods of the two hypotheses above for the counts for s, v and sv attested in the BNC, are:

$$L(H_1) = b(sv; v, p)b(s - sv; V - v; p) \tag{2}$$

$$L(H_2) = b(sv; v, p_1)b(s - sv; V - v; p_2) \tag{3}$$

The log of the likelihood ratio can then be computed as follows:

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)} \tag{4}$$

$$\log \lambda = \log \frac{b(sv; v, p)b(s - sv; V - v; p)}{b(sv; v, p_1)b(s - sv; V - v; p_2)} \tag{5}$$

$$\log \lambda = \log L(sv; v, p) + \log L(s - sv; V - v; p) \tag{6}$$

$$- \log L(sv; v, p_1) \tag{7}$$

$$- \log L(s - sv; V - v; p_2)$$

where: $L(k, n, x)$ is equal to $x^k(1 - x)^{n-k}$.

The collected data was presented to the machine learning algorithm as follows: for each verb, we have only two features. The first feature is the *n most likely head nouns in the subject position* of the verb, and the second feature is the *n most likely head nouns in the object position of the verb*. The variable n ranged from 5 to 25, in steps of 5. With *n most likely* we actually mean *the at most n most likely* subjects or objects. If we only find 10 different head nouns in the subject or object position of some verb, we still include it in our experiments where the variable n is larger than 10.

We conclude this section with Table 2, in which we list some verbs with their 5 most likely (according to *likelihood ratio*) subjects and nouns, to illustrate how we presented our data to the machine learning algorithm RIPPER. Table 2 also shows that verbs from the same semantic class (can) have some nouns in common in their list of most likely subjects or objects.

[1]The shallow parser was developed in co-operation with the ILK research group from the University of Tilburg (The Netherlands).

class	# verbs	%	subclass	# verbs	%	examples
18	70	21.3%	18.1	24	7.3%	*beat knock*
			18.2	11	3.4%	*bite shoot*
			18.3	25	7.6%	*flog belt*
			18.4	10	3.0%	*crash thud*
19	6	1.8%	/	/	/	*poke stick*
20	12	3.7%	/	/	/	*kiss lick*
21	42	12.8%	21.1	10	3.0%	*hack slash*
			21.2	32	9.8%	*chop squash*
22	142	43.3%	22.1	15	4.6%	*blend link*
			22.2	42	12.8%	*unify pair*
			22.3	29	8.8%	*roll splice*
			22.4	53	16.2%	*string knot*
			22.5	3	0.9%	*cleave cling*
23	56	17.1%	23.1	12	3.7%	*divide part*
			23.2	13	4.0%	*break pull*
			23.3	29	8.8%	*unzip unlace*
			23.4	2	0.6%	*differ diverge*

Table 1: The distribution of the verbs over the (sub)classes

verb	5 most likely subjects	5 most likely objects	main class label
pound	heart head foot rain blood	pavement stair earth road head	CLASS_18
drum	finger heart rain roar blood	finger business support interest heel	CLASS_18
chop	tbsp onion stir mushroom wash	parsley onion tomato garlic herb	CLASS_21
slice	blade onion oz carrot pain	bread tomato onion mushroom loaf	CLASS_21
seal	fate police lip door end	fate envelope victory gap deal	CLASS_22
clamp	hand finger car police mouth	hand teeth lip technique jaw	CLASS_22

Table 2: Some examples of verbs with their 5 most likely subjects or objects.

set-size	default setting	best setting	default baseline	random baseline
5	51.6	53.8	43.3	29.8
10	54.5	56.7		
15	53.4	54.2		
20	51.3	57.8		
25	52.7	56.7		

Table 3: Coarse-grained evaluation results – accuracy in percentages

4. Machine Learning Experiments

The machine learning algorithm we have experimented with is called RIPPER. RIPPER is an inductive rule learner: it induces classification rules from labeled examples by iteratively growing and then pruning rules. For more details on this algorithm, we refer to (Cohen, 1995) and (Cohen, 1996).

The advantage of using RIPPER is that it allows set-valued attributes: you do not need to convert the set-valued features to a binary format. Set-valued attributes is exactly what we are using: the feature *n most likely subjects* is the *set* of nouns appearing as head of the subject.

For each value of n, we searched the optimal parameter setting for this machine learning algorithm by doing leave-one-out training and testing: each one of the 277 verbs acted as test material, while the remaining 276 verbs were used as training material.

Depending on the type of features used – nominal, numeric, set-valued – RIPPER learns rules of the form "*if value for feature X (matches|contains|is greater than|is lesser than|...), then assign class label Y*". Below are two examples of rules – related to the verbs in Table 2 – learned by RIPPER from our dataset:

- `CLASS_21 4 0 IF OBJS ~ onion .`
- `CLASS_18 5 1 IF SUBJS ~ heart .`

We use nominal set-valued features, so these rules must be interpreted as "*if the set of n most likely objects contains onion, then assign class label CLASS_21*", and "*if the set of n most likely subjects contains heart, then assign class label CLASS_18*", respectively[2].

5. Results and Analysis

Table 3 shows the classification results of RIPPER, evaluated in the coarse-grained way. The numbers are accuracies expressed in percentages. The column *set-size* indicates the number of most likely subjects or objects we have used in the set-valued attributes for each verb. Though the accuracies are not very high, in all cases the default setting

[2]The two pairs of numbers in these rules (`4 0` and `5 1`) indicate the number of data points in the training set to which the rule applies: the first number in the pair is the number of data points for which the rule predicts the class correctly, the second number is the number of data points to which the rule assigns an incorrect class label.

set-size	default setting	best setting	default baseline	random baseline
5	23.1	25.6	16.2	9.4
10	26.7	28.5		
15	25.6	31.4		
20	24.6	31.1		
25	23.1	30.3		

Table 4: Fine-grained evaluation results – accuracy in percentages.

	18	19	20	21	22	23
prec.	58.5	0.0	66.7	75.0	57.4	33.3
rec.	45.3	0.0	72.7	25.0	90.6	7.0
$F_{B=1}$	55.3	/	67.8	53.6	62.0	19.0

Table 5: Precision, recall and $F_{B=1}$ scores for the six main classes.

scores better than both baseline results. With parameter optimization, we can improve the results a bit: the best result is obtained when the set-size is 20, yielding a classification accuracy of 58%.

Table 4 shows the results of RIPPER when analyzed in a fine-grained manner. It is clear that this task is much more difficult – but again all results with RIPPER's default settings are better than both baseline results. After parameter optimization and with a set-size of 15, the highest accuracy obtained is 31%.

In both evaluation types, the results are better than the baseline results, though the error reduction in the coarse-grained case is higher than in the fine-grained case. In the coarse-grained evaluation, the error reduction compared to the default baseline result is 23.6% and to the random baseline result is 39.8%. In the fine-grained case, the error reduction is 17.7% compared to the default and 24.3% compared to the random baseline.

Table 5 shows the precision, recall and $F_{B=1}$ scores for the six main classes in the best output we obtained with RIPPER in the coarse-grained evaluation. For most classes, precision is acceptable, but recall is quite low – exceptions are class 19 and 23. The reasonable precision but low recall suggests that for most classes, RIPPER learns a few rules which work well for a small set of verbs, but not for the whole class. The results for class 19 are very bad: it has zero precision and recall. Containing only 6 verbs, this class is the smallest: RIPPER does not have a lot of training material for this class. If we leave out during evaluation the classes with fewer than 10 verbs, which are class 19, 22.5 and 23.4, the classification accuracy improves a bit: 59% in the coarse-grained and 33% in the fine-grained case.

For class 22, recall is very high: more than 90%. This is because it is the *default class* for RIPPER: the machine learning algorithm starts by making rules for the smallest class first, then for the second smallest, and so on. For the largest class, there are no rules: if a new verb has to be classified, and all rules fail, RIPPER assigns it the label of the majority class.

The results in Tables 3, 4 and 5 indicate that to a certain extent, we can predict semantic classes from text with a machine learning algorithm by using little information provided by a shallow parser. For the coarse-grained case the results are reasonable, but for the fine-grained case we probably need more or other features.

6. Related Work

Table 6 summarizes very briefly the work of other researchers in the area of verb classification. The main difference between our research and the work summarized in Table 6 is that we have used nominal values, selected with a statistical criterion, whereas other researches have used numeric values – frequencies or probabilities.

The most work has been done by Merlo and Stevenson (see (Merlo and Stevenson, 2001; Stevenson and Merlo, 1999; Stevenson et al., 1999; Stevenson and Merlo, 2000; Merlo et al., 2002): with a decision tree learner and with frequency counts for five features, they obtain 69% classification accuracy. However, they classify verbs in only three classes which are not really semantically coherent and which do not correspond to classes from Beth Levin's classification.

In further research, Stevenson, in joint work with Joanis (Stevenson and Joanis, 2003), did use Levin's classes to evaluate the verb classification results: using a feature selection algorithm, which has to select among 220 features, and a decision tree learner, the best result they obtain is 58%. They also experimented with unsupervised learning, but results are much lower: their hierarchical clustering algorithm is able to reconstruct Levin's classification with 29% accuracy.

The state-of-the-art research comes from Schulte im Walde (Schulte im Walde, 1998): using frequency counts of verbs for a set of sub-categorization frames, she is able to reconstruct Levin's classification with unsupervised machine learning algorithms with 61% accuracy. She also did classification experiments with German verbs, using similar sub-categorization information (Schulte im Walde and Brew, 2002), but unfortunately she did not report the results in terms of classification accuracies.

Making a sound comparison of our results with the above mentioned research is not easy: they all use different classes and different machine learning methods. Moreover, it is never very clear whether the reported results are at the coarse-grained or at the fine-grained level. Still, we feel that our research can be best compared with Stevenson and Joanis' research – we even obtain similar results, 58% accuracy.

7. Future Work

In the following paragraphs we will briefly discuss our plans for near future work within the field of verb classification.

Comparison with other work. First of all, to make a sound comparison with other researchers' results, we will do similar experiments using the verbs used in Stevenson and Joanis' experiments (Stevenson and Joanis, 2003). The classes to which these verbs belong are listed in Table 7.

authors	classes	features	algorithm	result
Merlo and Stevenson	3 (Levin classes)	freq. counts for 5 features	C5.0	69%
Joanis and Stevenson	13 Levin classes	freq. counts for 220 features	C5.0	58%
			hierarchical clustering	29%
Schulte im Walde	30 Levin classes	freq. of verb with sub-categorization frames	iterative clustering	61%
			latent class analysis	54%

Table 6: A summary of related work.

class	subclasses
9	9.1-6 (other verbs of putting) 9.7 (spray/load verbs) 9.8 (fill verbs)
10	10.1, 10.5 (steal and remove verbs) 10.4.1-2 (wipe verbs) 10.6 (cheat verbs)
13	13.1, 13.3 (recipient verbs)
26	26.1, 26.3 (benefactive verbs) 26.1, 26.3, 26.7 (object-drop verbs)
31	31.1 (amuse verbs) 32.2 (admire verbs)
43	43.2 (sound emission verbs)
45	45.1-4 (change of state verbs)
51	51.3.2 (run verbs)

Table 7: Levin's classes used in Stevenson and Joanis' experiments.

The class labels between brackets in this table are Stevenson and Joanis' interpretation of Levin's classes. The granularity of this classification is somewhere in between what we've called coarse- and fine-grained.

More features. We will also try to add more features which a shallow parser can provide, like for example the prepositions following a verb and the list of nouns in the prepositional phrase, and do similar experiments to find out whether these features can contribute to verb classification.

Token-based verb classification. Our verb classification experiments reported in this paper were *type-based*: information is collected by looking at individual tokens of a verb in a corpus, and for each verb, this information was collapsed in one data vector. It is interesting to investigate whether a *token-based* approach will also be successful at classifying verbs. The experimental set-up will then be as follows: for each token of a verb in a set of n verbs, a vector with information from a shallow parsed corpus (nominal values such as Part-of-Speech, chunk and relation tags of the focus word and surrounding words) will be constructed. For testing/evaluating this approach, we will do some kind of *leave-one-out cross-validation*: we will use all vectors for the tokens of n-1 verbs as training material, and classify all vectors for the tokens of the remaining verb (the *unknown* verb). In this architecture, the semantic class of the *unknown verb* is the label that is most often predicted.

This work is planned for the near future, and the results will be presented and discussed at the workshop.

Acknowledgments

This research was carried out within the context of the SemaDuct project, sponsored by the *Fonds voor Wetenschappelijk Onderzoek – Vlaanderen* (Fund for Scientific Research – Flanders).

8. References

Buchholz, Sabine, Jorn Veenstra, and Walter Daelemans, 1999. Cascaded grammatical relation assignment. In *Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing*. University of Maryland, College Park, MD, USA: The Association for Computational Linguistics.

Cohen, William W., 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*. Tahoe City, CA, USA.

Cohen, William W., 1996. Learning with set-valued features. In *Proceedings of the 13th National Conference on Artificial Intelligence*. Portland, Oregon, USA: The American Association for Artificial Intelligence.

Daelemans, Walter, Sabine Buchholz, and Jorn Veenstra, 1999. Memory-based shallow parsing. In *Proceedings of the Conference on Natural Language Learning*. Bergen, Norway: The Association for Computational Linguistics.

Levin, Beth, 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago, IL, USA: The University of Chicago Press.

Manning, Christopher D. and Hinrich Schütze, 1999. *Foundations of Statistical Natural Language Processing*, chapter 8. Cambridge, MA, USA: The MIT Press, 2nd edition.

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

Merlo, Paola, Suzanne Stevenson, Vivian Tsang, and Gianluca Allaria, 2002. A multilingual paradigm for automatic verb classification. In (The Association for Computational Linguistics, 2002), pages 207–214.

Schulte im Walde, Sabine, 1998. Automatic semantic classification of verbs according to their alternation behaviour. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung*, 4(3):55–96. Lehrstuhl für Theoretische Computerlinguistik, Universität Stuttgart.

Schulte im Walde, Sabine and Chris Brew, 2002. Inducing german semantic verb classes from purely syntactic subcategorisation information. In (The Association for Computational Linguistics, 2002), pages 223–230.

Stevenson, Suzanne and Eric Joanis, 2003. Semi-supervised verb class discovery using noisy features. In *Proceedings of the Conference on Natural Language Learning*. Edmonton, Canada: The Association for Computational Linguistics.

Stevenson, Suzanne and Paola Merlo, 1999. Automatic verb classification using grammatical features. In *Proceedings of the 9th Conference of The European Chapter of The Association for Computational Linguistics*. Bergen, Norway: The Association for Computational Linguistics.

Stevenson, Suzanne and Paola Merlo, 2000. Automatic lexical acquisition based on statistical distributions. In *Proceedings of the 18th International Conference on Computational Linguistics*. Saarland University, Saarbrücken, Germany: International Committee on Computational Linguistics.

Stevenson, Suzanne, Paola Merlo, Natalia Kariaeva, and Kamin Whitehouse, 1999. Supervised learning of lexical semantic verb classes using frequency distributions. In *Proceedings of the SIGLEX-99 Workshop: Standardizing Lexical Resources*. University of Maryland, College Park, MD, USA: The Association for Computational Linguistics.

The Association for Computational Linguistics, 2002. *Proceedings of the 40th Annual Meeting of The Association for Computational Linguistics*. University of Pennsylvania, Philadelphia, PA, USA.