Automatically Deriving Tectogrammatical Labels from Other Resources

A Comparison of Semantic Labels Across Frameworks

Owen Rambow Columbia U.
Bonnie Dorr U. of Maryland
Karin Kipper U. of Penn.
Ivona Kučerová MIT
Martha Palmer U. of Penn.
rambow@cs.columbia.edu

Abstract

Local semantic labels are relevant to the verb meaning in question, while *global* semantic labels are relevant across different verbs and verb meanings. We show that global semantic labels from different frameworks are not necessarily compatible and argue that, therefore, corpora should be annotated with local semantic labels and that global semantic labels should then be automatically annotated using framework-specific lexicons.

1 Introduction

The development of the Penn Treebank (PTB) (Marcus et al., 1993; Marcus et al., 1994) has had an immense effect on the development of natural language processing (NLP) by providing training and testing data for approaches based on machine learning, including statistical models. It has inspired other treebanking efforts in many languages, including the Prague Dependency Treebank (PDTB) (Böhmová et al., 2001; Hajic et al., 2001). However, since the development of the PTB, it has become clear that for many NLP applications, parsing to a a level of representation is needed that is "deeper" than the surface-syntactic phrase structure of the PTB. Furthermore, work in generation using machine learning cannot use the PTB because the representation is too shallow as a starting point for generation. Thus, a more richly annotated corpus is needed, in particular, a corpus that includes certain semantic notions. Annotation efforts for languages other than English have been able to incorporate this requirement from the beginning. For example, the PDTB includes both the Analytical and the Tectogrammatical level of representation. However, for English, such resources have been created only recently . One such resource is the PropBank (Kingsbury et al., 2002), which superimposes an annotation for verbal predicates and their arguments and adjuncts on the PTB. In the PropBank, the annotation of the relation between verb and dependent is "local", i.e., only relevant to a single verb meaning. However, for many applications we need a "global" semantic labeling scheme such as that provided by the Tectogrammatical Representation (TR) of the PDTB, with labels such as ACT (actor) and ORIG (origin) whose meaning is not specific to the verb. The question arises whether and how the PropBank can be extended to reflect global semantic information.

The direct motivation for this paper is the observation by Hajičová and Kučerová (2002) that the global semantics of the Tectogrammatical Representation (TR) of the Prague school cannot be derived directly from the local semantics of the PropBank, since it does not contain sufficient detail: TR makes distinctions not made in the PropBank. The authors suggest that it may, however, be derivable from the PropBank with the aid of an intermediary representation that also uses global semantic labels such as Lexical-Conceptual Structure (LCS), or VerbNet (VN). The proposal is worth investigating: it seems reasonable to derive TR labels from other representations of global semantics. While TR, LCS, and VN use different labels, we expect

there to be some consistency. For example, LCS src should correspond to VerbNet Source and TR ORIG. While the three representations — TR, LCS, VN — are based on different approaches to representing the meaning of a sentence, all three approaches assume that there is a sharable semantic intuition about the meaning of the relation between a verb and each of its dependents (argument or adjunct). Of course, the semantic labels themselves differ (as in the case of src, Source, and ORIG), and furthermore, often one approach makes finer-grained distinctions than another, for example VN has one category Time, while TR has many subcategories, including THL (temporal length) and THWHEN (time point) and so on. Nonetheless, in these cases, the different label sets are compatible in meaning, in the sense that we can define a one-to-many mapping between label sets in the different frameworks. More precisely, we expect one of three situations to hold for a given pair of labels from label sets $\mathcal{A}$ and $\mathcal{B}$:

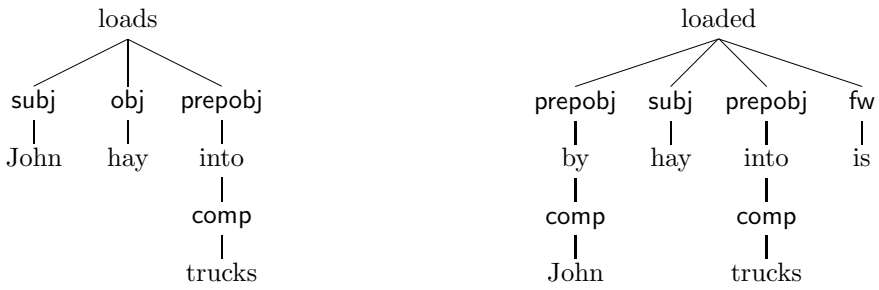
- A label a in $\mathcal{A}$ corresponds to exactly one label b in $\mathcal{B}$, and b corresponds only to a (bijective case).
- A label a in $\mathcal{A}$ corresponds to a set of labels B in $\mathcal{B}$, and each element b of B corresponds only to a (one-to-many case).
- A label b in $\mathcal{B}$ corresponds to a set of labels A in $\mathcal{A}$, and each element a of A corresponds only to b (many-to-one case).

The case in which there are overlapping meanings, with a_1 from $\mathcal{A}$ corresponding to b_1 and b_2 from $\mathcal{B}$, and a_2 from $\mathcal{A}$ corresponding to b_2 and b_3 from $\mathcal{B}$, should be excluded.

There are two positions one may take. Given that global semantic labels express relationships which are meaningful across verbs, and assuming that researchers in different frameworks share certain semantic intuitions, we may claim that labels are (possibly) compatible across frameworks. On the other hand, we may claim that in such difficult semantic issues, it is unreasonable to expect different frameworks to have converged on label sets with compatible meanings. The issue is not just one of academic interest — it is also of great practical interest. If the usefulness of parsing is to be increased by developing semantically annotated corpora (a very costly process), it is important to know whether an annotation in, for example, LCS will allow us to automatically derive a corpus annotated in TR. If not, the value of a corpus of LCS labels will be reduced, since it will be relevant to a smaller community of researchers (those working in the framework of LCS). While to some researchers the answer to the question of inter-framework compatibility of labels may be intuitively obvious, we are not aware of any serious empirical study of this question. Such a study must necessarily be corpus-based or experimental, as only the data will reveal how the frameworks *use* their labels (as opposed to defining them), which is what this question is about.

In this paper, we present the results of investigating the relationship between PropBank, TR, LCS, and VN labels based on an annotated corpus. The conclusion we put forward is that global semantic labels are not only framework-specific, but also lexically idiosyncratic *within* each framework. This means that labels are not compatible between frameworks, and do not necessarily express the same semantic intuition. (It of course does not mean that these labels are used inconsistently within any one framework.) As a result, we argue that corpora should not be annotated in terms of global semantic labels (such as TR, LCS, or VN). Instead, we argue that corpora should be annotated with local semantic labels (as has already been done in the PropBank), and global semantic labels should be generated automatically using framework-specific lexicons (i.e., verb-specific lists of label mappings for arguments). Such lexicons represent an important resource in their own right.

This paper is structured as follows. We start by introducing a vocabulary to talk about types of resources in general in Section 2. We then present four different ways of labeling corpora



John loads hay into trucks

Hay is loaded into trucks by John

Figure 1: Surface syntactic representation for the sentences in (1)

with semantic information: PropBank in Section 3, TR in Section 4, VerbNet in Section 5, and LCS in Section 6.[1] While these approached are part of larger theories of syntax and lexical semantics, we are for the purpose of this paper only interested in the label set they use to annotate the relation between a verbal predicate and its arguments and adjuncts; we will therefore refer to these theories in a reductive manner as "labeling schemes". We then compare the global-semantic labeling schemes to each other in Section 7 and find labeling to be lexically idiosyncratic and framework-specific. In Section 8 we return to the original question of Hajičová and Kučerová (2002) and report on experiments using machine learning to derive rule sets for annotating a corpus with TR labels. These results confirm the conclusions of Section 7.

2 Types of Corpus Labels

Surface syntax reflects the relation between words at the surface level. Consider the following pair of sentences, whose structure is shown in Figure 1:

(1) a. John loads hay into trucks
    b. Hay is loaded into trucks by John

In this example, where two sentences differ only in the voice of the verb, the first two arguments of the verb, *John* and *hay*, have different roles depending on voice. The (dependency representation recoverable from the) PTB has a surface-syntactic labeling scheme, though deeper labels can be inferred from tags and traces.

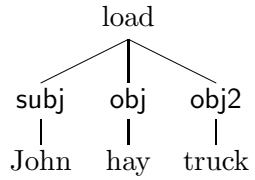
Deep syntax normalizes syntactically productive alternations (those that apply to all or a well-defined class of verbs, not lexically idiosyncratically). This primarily refers to voice, but (perhaps) also other transformations such as dative shift. The deep-syntactic representation for the two trees in Figure 1 (i.e., the two sentences in (1) is shown in Figure 2. However, the deep-syntactic representation does not capture verb-specific alternations, such as the container-content alternation found with *load*:

(2) a. John loads hay into trucks

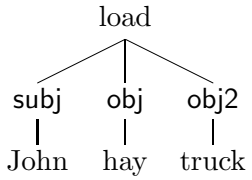
    b. John loads trucks with hay

In these two sentences, the semantic relationship between the verb and its three arguments is the same in both sentences, but they are realized differently syntactically: *hay* is the deep direct object in one, *trucks* in the other. This is shown in the two trees in Figure 3.

[1] These labeling schemes in themselves are not the original work presented in this paper, we summarize them here for the convenience of the reader. The original work is investigating the relation between and among them.

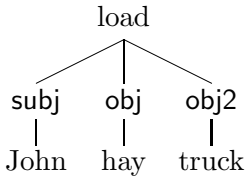


John loads hay into trucks

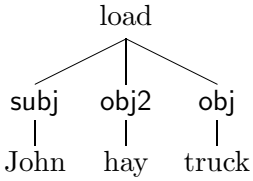


Hay is loaded into trucks by Johnny

Figure 2: Deep-syntactic representation



John loads hay into trucks



John loads trucks with hay

Figure 3: Deep-syntactic representation: missed generalization

Instead, we can choose numerical labels (arg0, arg1, ...) on the arguments which abstract away from the syntactic realization and only represent the semantic relation between the particular verb meaning and the arguments. These **local semantic** labels have no intrinsic meaning and are significant only when several syntactic realizations of the same verb meaning are contrasted. An example is shown in Figure 4.

Now consider the following two sentences:

(3) a. John loads hay into trucks

b. John throws hay into trucks

Semantically, one could claim that (3b) merely adds manner information to (3a), and that therefore the arguments should have the same relationships to the verb in the two cases. However, since these are different verbs (and *a fortiori* different verb meanings) there is no guarantee that the local semantic arc labels are the same. In a **global semantic** annotation, the arc labels do not reflect syntax at all, and are meaningful across verbs and verb meanings. The

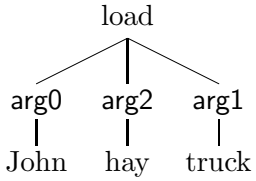


Figure 4: Local semantic representation for both (2a) and (2b)

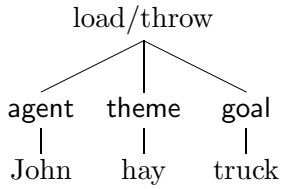


Figure 5: Global semantic representation for (3a) (with *load*) and (3b) (with*throw*); the labels used are for illustrative purposes

labels reflect generalizations about the types of relations that can exist between a verb and its argument, and the representation in Figure 5 applies to sentences (3a) and (3b).[2]

3 PropBank

The PropBank (Kingsbury et al., 2002) annotates the Penn Wall Street Journal Treebank II with dependency structures (or 'predicate-argument' structures), using sense tags for each word and local semantic labels for each argument and adjunct. The argument labels are numbered and used consistently across syntactic alternations for the same verb meaning, as shown in Figure 4. Adjuncts are given special tags such as TMP (for temporal), or LOC (for locatives) derived from the original annotation of the Penn Treebank. In addition to the annotated corpus, PropBank provides a lexicon which lists, for each meaning of each annotated verb, its *roleset*, i.e., the possible arguments in the predicate and their labels. An example, the entry for the verb *kick*, is given in Figure 6. The notion of "meaning" used is fairly coarse-grained, and it is typically motivated from differing syntactic behavior. Since each verb meaning corresponds to exactly one roleset, these terms are often used interchangeably. The roleset also includes a "descriptor" field which is intended for use during annotation and as documentation, but which does not have any theoretical standing. Each entry also includes examples. Currently there are frames for about 1600 verbs in the corpus, with a total of 2402 rolesets.

ID	kick.01	
Name	drive or impel with the foot	
VN/Levin classes	11.4-2, 17.1, 18.1, 23.2 40.3.2, 49	
Roles	Number	Description
	0	Kicker
	1	Thing kicked
	2	Instrument (defaults to foot)
Example	$[\text{John}]_i$ tried $[\text{*trace*}_i]_{\text{ARG0}}$ to kick $[\text{the football}]_{\text{ARG1}}$	

Figure 6: The unique roleset for *kick*

[2]The FrameNet project (Baker et al., 1998) uses semantic labels which are local, but apply not to one verb meaning, but to a set of verb meanings that refer to the same frame (i.e., situation). For example, *buy*, *sell*, *cost* and so on all refer to the commercial transaction frame, realizing different participants of the frame in different syntactic ways. However, since the frame elements such as Buyer or Rate (=price) do not refer to an abstract notion of the relationship between a proposition and its argument, but rather to a specific set of verbs and a specific argument, the approach is closer in spirit to a local semantic approach. Perhaps a better term for FrameNet would be "regional semantics".

PropBank		TR	
Role	Description	Form	label
0	Bidder	subject	ACT
1	Target	*for*	EFF
		to	AIM
2	Amount bid	object	PAT

Figure 7: TR extension to PropBank entry for *bid*, roleset name "auction"

4 Tectogrammatical Representation

The Tectogrammatical Representation (TR) of the Prague School (Sgall et al., 1986) is a dependency representation that contains only autosemantic (=meaning-bearing) words. The arcs are labeled with rich set of labels. What distinguishes TR from other labeling schemes is that it is hybrid: the deep subject and deep object of a verb are always given the labels ACT (for actor) and PAT (for patient), respectively. The deep indirect object is given one of three labels, EFF(ect), ADDR(essee), or ORIG(in). Other arguments and free adjuncts are drawn from list of 42 global semantic labels, such as AIM, BEN(eficiary), LOC(ation), MAN(ner), and a large number of temporal adjuncts such as THL (temporal length) and THWHEN (time point).

For the TR, we have a small gold standard. Approximately 1,500 sentences of the PTB were annotated with TR dependency structure and arc labels. A total of 36 different labels are used in this corpus. The sentences were prepared automatically by a computer program (Žabokrtský and Kučerová, 2002) and then corrected manually. We will refer to this corpus as the TRGS (which should not be confused with the PDTB, which is a much larger corpus in Czech), and to the code as AutoTR. It uses heuristics that can access the full PTB annotation.

In addition, there is a lexicon of tectogrammatical entries for English based on (a subset of) the PropBank lexicon. The mapping was done for 662 predicates (all PropBank entries that were done by January 2002). Every entry contains an original PropBank lexical information with examples, information about Levin class membership and appropriate tectogrammatical mapping. The mapping is only defined for entries that are explicitly listed in the original PropBank entry; no others were created. Figure 7 shows the entry for the verb *bid*. Note that the mapping to TR is indexed both on the Propbank argument and on the syntactic realization ("form"), so that arg1 may become EFF or AIM, depending on the preposition that it is realized with.

We evaluated the quality of the PropBank-to-TR lexicon by comparing results on those arguments in the TRGS whose verbs are also in the lexicon (727 argument instances). The AutoTR program has an error rate of 15.3% on this data, while the lexicon's error rate is only 12.2%. We performed an error analysis by randomly sampling 15 instances (of the 89 errors). In 9 instances, there were inconsistencies between the lexicon and the TRGS. (Of these, one instance was furthermore inconsistent in the TRGS.) In four instances, there appeared to be an error in the lexicon. And in two instances, there was an error in our automatic alignment of the data due to a mismatch of the syntactic analysis in the TRGS and in the PTB. We conclude that all these problems are in principle fixable.

5 VerbNet

VerbNet (Kipper et al., 2000) is a hierarchical verb lexicon with syntactic and semantic information for English verbs, using Levin verb classes (Levin, 1993) to systematically construct lexical entries. The first level in the hierarchy is constituted by the original Levin classes, with each class subsequently refined to account for further semantic and syntactic differences within

Actor, Agent, Theme, Patient, Asset, Attribute, Beneficiary, Cause, Destination, Experiencer, Instrument, Location, Material, Patient, Product, Recipient, Source, Stimulus, Time, Topic

Figure 8: Inventory of thematic role labels used in VerbNet

a class. Each node in the hierarchy is characterized extensionally by its set of verbs, and intensionally by a list of the arguments of those verbs and syntactic and semantic information about the verbs. The argument list consists of thematic labels from a set of 20 possible such labels (given in Fig. 8), and possibly selectional restrictions on the arguments expressed using binary predicates. The syntactic information maps the list of thematic arguments to deep-syntactic arguments. The semantic information for the verbs is expressed as a set (i.e., conjunction) of semantic predicates, such as *motion, contact, transfer_info.*[3] Currently, all Levin verb classes have been assigned thematic roles and syntactic frames, and 123 classes, with more than 2500 verbs, are completely described, including their semantic predicates.

In addition, a PropBank-to-VerbNet lexicon maps the rolesets of PropBank to VerbNet classes, and also the PropBank argument labels in the rolesets to VerbNet thematic role labels. Fig. 9 shows an example of the mapping of roleset *install.01* with VerbNet class *put-9.1*. The mapping is currently not complete: some verb meanings in PropBank have not yet been mapped, others are mapped to several VerbNet classes as the PropBank verb meanings are sometimes coarser than or simply different from the VerbNet verb meanings (many PropBank rolesets are based on a financial corpus and have a very specific meaning).

PropBank		VN
Role	Description	label
0	Putter	Agent
1	Thing put	Theme
2	Where put	Destination
VerbNet-Levin class		9.1

Figure 9: Entry in PropBank-to-VerbNet lexicon for *put* (excerpt)

Using this lexicon, we have augmented the PropBank-annotated Penn Treebank with VerbNet annotations automatically. In theory, we could simply look up the corresponding VerbNet argument for each annotated PropBank argument in the corpus. However, there are several impediments to doing this. First, the PropBank annotation of the Penn Treebank does not currently include the roleset, i.e., the verb meaning: of all the PropBank-annotated verbs in the TRGS, in only 74.7% of cases do we have access to the PropBank meaning (roleset). Second, because the PropBank-to-VerbNet lexicon is not yet complete (as just described), only 42.1% of verbs (instances) have exactly one VerbNet-Levin class assigned to them. Therefore, only 46.1% of argument instances can be assigned VerbNet thematic roles automatically (18 different labels are used) However, the coverage will increase as (i) PropBank annotates rolesets in the corpus and (ii) the annotation of the PropBank lexicon with VerbNet information progresses. In principle, there is no reason why we cannot achieve a near 100% automatic coverage of the hand-annotated PropBank arguments in the Penn Treebank with VerbNet thematic roles.

[3]Both for VerbNet and LCS, the semantic information about each verb is not directly germane to this paper.

Verb	jog
Class	51.3.2.a.ii
Theta	_th,src(),goal()

Figure 10: LCS definitions of *jog* (excerpt)

6 Lexical Conceptual Structure

Lexical Conceptual Structure (LCS) is a compositional abstraction with language-independent properties that transcend structural idiosyncrasies (Jackendoff, 1983; Dorr, 1997). LCS captures the semantics of a lexical item through a combination of semantic structure (specified by the shape of the graph and its structural primitives and fields) and semantic content (specified through constants). The semantic structure of a verb is something the verb inherits from its Levin verb class, whereas the content comes from the specific verb itself.

The lexicon entry for one sense of the English verb *jog* is shown in Figure 10. This entry includes several pieces of information such as the word's semantic verb class, its thematic roles ("Theta" – in this case, th, src, and goal), and the LCS itself (not shown here, as it is not directly relevant to this paper). The LCS specifies how the arguments — identified by their thematic roles — contribute to the meaning of the verb.

Figure 11 contains a list of thematic roles. The theta-role specification indicates the obligatory and optional roles by an underscore (_) and a comma (,), respectively. The roles are ordered in a canonical order normalized for voice (and dative shift): subject; object; indirect object; etc, which corresponds to surface order in English. Thus, the _th_loc grid is not the same as the _loc_th grid (*The box holds the ball* as opposed to *The water fills the box*).

agent, theme, experiencer, information, src (source), goal, perceived item, pred (identificational predicate), locational predicate, mod-poss (possessed item modifier), mod-pred (property modifier)

Figure 11: Inventory of LCS thematic roles (extract)

To derive LCS thematic labels for arguments and adjuncts in the PropBank, we make use of the Lexical Verb Database (LVD). This resource contains hand-constructed LCSs organized into semantic classes — a reformulated version of the semantic classes in (Levin, 1993). The LVD contains 4432 verbs in 492 classes with more specific numbering than the original Levin numbering (e.g., "51.3.2.a.ii"), a total of 11000 verb entries. For the mapping, we used as keys into the LVD both the lexeme and the Levin class as determined by VerbNet (see Section 5), adjusting the class name to account for the different extensions developed by Verbnet and LCS. Each key returns a set of possible theta grids for each lookup. We then form the intersection of the two sets, to get at the theta grid for the verb in its specific meaning. If this intersection is empty, we instead form the union. (This complex approach maximizes coverage.) We then map to each argument a set of possible theta roles (note that even if there are two possible theta grids, one of the arguments may receive the same role under both). This approach yields 54.7% of verb instances in the TRGS with a unique theta-grid, and 47.7% of argument/adjunct instances, with a unique theta role. (The lower figure is presumably due to the fact that verbs with fewer arguments are more likely to have unique theta grids.) A total of 13 LCS roles are used for these instances.

Predict	From	No mlex	With mlex	n
VN	LCS	30.3%	13.0%	399
LCS	VN	22.8%	9.5%	399
TR	VN	36.1%	14.4%	97
VN	TR	56.7%	8.3%	97
TR	LCS	42.3%	20.5%	78
LCS	TR	41.0%	11.5%	78

Figure 12: Error rates for predicting one label set from another, with and without using feature mlex (the governing verb's lexeme); n is the number of tokens for the study

VN label	TR label	tokens	types	sample verbs
Topic	EFF	29	2	say X
Predicate	EFF	12	7	view Y as X
	AIM	2	2	use Y to do X
	CPR	1	1	rank Y as X
	COMPL	1	1	believe Y that X
	LOC	1	1	engage Y in X
Attribute	EFF	4	3	rate Y X
	EXT	1	1	last X
	THL	1	1	last X
	DIFF	1	1	fall X
	LOC	1	1	price Y at X

Figure 13: Exhaustive mapping of three VerbNet labels to TR labels other than ACT and PAT (the argument being labeled is X)

7 Relation Between Semantic Labels

We now address the question of how similar the three annotation schemes are, i.e., the semantic part of TR, LCS, and VerbNet. To test the correspondence between global semantic labels, we use Ripper (Cohen, 1996) to predict one label set, given another. Using a set of attributes, Ripper greedily learns rule sets that choose one of several classes for each data set. Because in this section we are using Ripper to analyze the data, not to actually learn rule sets to apply to unseen data (as we do in Section 8), we report here the error rate on the training data.

For these experiments, we use all arguments from the TRGS which are also labeled in the PropBank, 1268 data points. For VN and LCS, we exclude all data points in which either the predictor label or the predicted label are not available from the mappings described in Sections 4, 5, and 6, respectively. In the case of TR (which is always available), we exclude cases with the ACT and PAT features, as they are determined syntactically. If there is a one-to-one correspondence between two label sets, we expect a zero error rate for both directions; if the correspondence is one-to-many (i.e., one label set is more detailed than the other), we expect a zero error rate for at least one direction.

Instead, what we find are error rates between 22.8% and 56.7%, for all directions. Crucially, we find these error rates greatly reduced (with error reduction ranging between 51% and 85%) if we also allow the lexeme of the governing verb to be a feature. The results are summarized in Figure 12. All differences are significant, using the usual Ripper test (the difference between the results must be larger than twice the sum of each run's standard deviation). As expected, in each pair, the richer label set (as measured by the number of labels used in the TRGS) is better at predicting the less rich label set.

By way of illustration, we will look in more detail at the way in which three VN labels, Topic, Predicate, and Attribute, map to TR categories. The data is summarized in Figure 13.[4] As we can see, for all three labels, the most common TR label (and in the case of Topic, the only TR label) is EFF. However, closer inspection reveals this not to be the case. VerbNet makes a distinction between the communicated content (*John said he is happy*) which is a Topic, a Predicate of another dependent of the verb (*they view/portray/describe the sales force as a critical asset*, where *a critical asset* is a predicate true of the sales force), and an Attribute of another actant of the verb (*they value/estimate the order at $326 million/rate the bond AAA*).[5] TR considers all these cases to be EFFects of an act of communication or judgment. Conversely, TR makes a distinction between an EFFect of a human action (of communication or judgment, such *they value/estimate the order at $326 million/rate the bond AAA*) and different types of states of affairs, for example a DIFFerence (*UAL stock has fallen 33%*) or a length of time (THL, *the earth quake lasted 15 seconds*). To VN, these are all Attributes.

But note that in nearly all cases considered in the table in Figure 13, the governing verb determines the label assignment both for TR and VN.[6] Thus, both in the general Ripper experiments and in these specific examples, we see that there is no general mapping among the labels; instead, we must take the governing verb into account. We conclude that assigning labels is both framework specific and lexically idiosyncratic within each framework.

```
Final hypothesis is:
ORIG if fw=from and vn!=_ (2/1).
CAUS if fw=because (2/0).
COND if fw=if (3/0).
MOD if lemma=probably (2/0).
DIR3 if pb=ARG2 and pba=DIR (2/0).
AIM if fw=to and vrole=adj (12/4).
MANN if pba=MNR (20/1).
ADDR if pb=ARG2 and vn=Recipient and
  lemma!=blame and lemma!=article (7/0).
ADDR if lemma=audience (2/1).
ADDR if mlemma=assure and pb=ARG1 (2/0).
TWHEN if pba=TMP (55/6).
EFF if vn=Topic and mlemma=say (25/0).
EFF if vrole='2' and fw=as (12/1).
ACT if vrole='0' (366/16).
default PATC (502/67).
```

Figure 14: Sample generated rule set (excerpt — "fw" is the function word for the argument, "mlex" the governing verb's lexeme, "pba" the modifier tag from the Penn Treebank)

8 Predicting TR Labels

We now turn to experiments for learning rule sets for choosing TR labels from all other labels (the task described by Hajičová and Kučerová (2002), the original inspiration for this work). We again use Ripper, as in Section 7. The task is to predict the TR label, and we experiment with different feature sets. Given our analysis in Section 7, we predict that using other global

[4] We exclude tokens whose TR labels are ACT or PAT, as these labels are determined entirely syntactically.

[5] Intuitively, a *predicate* is a function from entities to truth values, while an *attribute* is a function from entities to an open set of possible values (such as dollar amounts).

[6] The exceptions are in TR: *use Y to do X* is sometimes EFF, sometimes AIM, while *last X* is sometimes EXTent, sometimes temporal length (THL). We assume these are labeling inconsistencies.

semantic labels, i.e., VN or LCS, will *not* improve performance. However, we expect syntactic (including lexical) features and local semantic features (PropBank) to contribute to performance. We observe that it is not clear what the topline is, given some inconsistency in the gold standard; the experience reported above from very small hand-inspected data sets suggests an inconsistency rate of between 5% and 10%.

We use the following syntactic features: **PTB lean** (argument lemma, governing verb's lemma, part-of-speech, and function word, if any); **Full PTB** (PTB lean + TR label of mother, extended tag of PTB, node labels of path to root); **VRole** (the deep-syntactic argument, as derived from the PTB by head percolation and voice normalization); and **AutoTR**, the computer script AutoTR writtem to determine TR labels. We also use these semantic features: PropBank, LCS, VerbNet. A sample rule set (with features PTB-lean, Vrole, Propbank, and VerbNet) is shown in Figure 14. The rules are checked from top to bottom, when one applies the listed label is chosen. The numbers in parentheses indicate the number of times the rule applies correctly (before the slash) and incorrectly (after the slash). Clearly, there is some overfitting happening in this particular ruleset (for example, in the rule to choose ADDR if the lemma is *audience*).

The results for the machine learning experiments are summarized in Figure 15. These are based on five-fold cross-validation on a set of 1268 data points (those arguments of the TRGS labeled by PropBank, with mismatches related to different syntactic treatment of conjunction removed). Note that because of the greedy nature of Ripper, a superset of features may (and often does) produce worse results than a subset. In general, any two results are statistically significant if their difference is between three and five; there are too many combinations to list all. Compared to the baseline of the hand-written AutoTR code, the combination of PTB Lean, Vrole, and PropBank provides an error reduction of 24.5% with respect to a (possibly unrealistic) 0% error topline. The error reduction is 75.8% with respect to default baseline of always choosing PAT, the most common label (i.e., running Ripper with no features), and the 0% topline.

Semantics Syntax	None	PropBank	PB&LCS	PB&VN
None	59.23%	24.30%	23.27%	22.25%
Vrole	30.44%	19.80%	18.38%	17.75%
PTB	18.15%	15.70%	16.17%	16.02%
PTB & Vrole	16.09%	15.14%	15.46%	14.67%
PTB Lean & Vrole	16.80%	14.36%	15.15%	14.51%

Figure 15: Results (error rate) for different combinations of syntactic features (left column) and semantic features (top row); baseline error rate using hand-written AutoTR code is 19.01%

We now highlight some important conclusions (all are statistically significant unless otherwise stated). First, some syntax always helps, whether or not we have semantics (compare the rows labeled "None" and any of the rows below it). This is not surprising, as some of the TR labels (ACT and PAT) are defined fully syntactically. Second, the PTB-lean feature set does as well as the full PTB set, no matter what semantic information is used (compare rows labeled "PTB & Vrole" and "PTB Lean & Vrole"). In particular, the TR label of mother, the extended tag of the PTB, and the node labels of path to root do not help. Third, using the PropBank improves on using just syntactic information (compare the columns labeled "None" and "PropBank" — not all pairwise comparisons are statistically significant). Fourth, as predicted, there is no benefit to adding global semantic information once local semantic information is used (compare the column labeled "PropBank" to the columns labeled "PB&LCS" and "PB&VN").

In related work, Gildea and Jurafsky (2002) predict generic FrameNet labels (similar to the VN or LCS labels). They achieve an error rate of 17.9% using no other semantic information. While this error rate is similar to the ones we report here (in the row labeled "None"), there are some important differences: their testing data only contains seen predicates (unlike ours), but our task is facilitated by the fact that the most common labels in TR are defined syntactically.

9 Conclusions

As we have seen, there are problems in mapping among VerbNet, LCS, and TR. Most truly global semantic labels are both framework-specific and lexically idiosyncratic: different frameworks (and possibly researchers in the same framework) do not divide up the space of possible labels in the same way. As a result, in automatically labeling a corpus with TR labels, using LCS or VerbNet does not improve on using only syntactic (including lexical) and local semantic information, contrary to the suggestion of Hajičová and Kučerová (2002).

For future efforts to label corpora with global semantic labels, we therefore suggest using a (framework-specific, lexically idiosyncratic) PropBank-to-X lexicon,[7] where "X" is the desired labeling scheme (which can be a very domain- or application-specific, for example specific to a particular information-extraction task). A PropBank-style corpus is easier to annotate than global semantics (as suggested by anecdotal evidence on inter-annotator agreement), and a PropBank-to-X lexicon is also easier to create than an annotated corpus. Hence, if you want global semantic labels in a corpus, annotate local semantics (or use an existing resource), and create a translation lexicon. In this way, two reusable resources are created (the corpus annotated in local semantics and the lexicon), rather than one non-reusable resource (a corpus labeled in a particular global semantic scheme).

References

Collin F. Baker, J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 86–90, Montréal.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2001. The Prague Dependency Treebank: Three-Level Annotation Scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers.

William Cohen. 1996. Learning trees and rules with set-valued features. In *Fourteenth Conference of the American Association of Artificial Intelligence*. AAAI.

Bonnie J. Dorr. 1997. Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation. *Machine Translation*, 12(4):271–322.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Jan Hajic, Eva Hajicová, Martin Holub, Petr Pajas, Petr Sgall, Barbora Vidová-Hladká, and Veronika eznícková. 2001. The current status of the prague dependency treebank. In *LNAI 2166*, LNAI 2166, pages 11–20. Springer Verlag, Berlin, Heidelberg, New York.

Eva Hajičová and Ivona Kučerová. 2002. Argument/valency structure in PropBank, LCS database and Prague Dependency Treebank: A comparative study. In *Proceedings of LREC*.

Ray Jackendoff. 1983. *Semantics and Cognition*. The MIT Press, Cambridge, MA.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology Conference*, San Diego, CA.

[7] "PropBank" can be replaced by any other local semantic labeling scheme.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, July-August.

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

Mitchell M. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19.2:313–330, June.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop*.

P. Sgall, E. Hajicova, and J. Panevova. 1986. *The meaning of the sentence and its semantic and pragmatic aspects*. Reidel, Dordrecht.

Zdeněk Žabokrtský and Ivona Kučerová. 2002. Transforming penn treebank phrase trees into (praguian) tectogrammatical dependency trees. *The Prague Bulletin of Mathematical Linguistics*, (78):77–102.