

Adapting Decision Trees for Learning Selectional Restrictions

Sean Szumlanski and Fernando Gomez

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
{seansz, gomez}@cs.ucf.edu

Abstract

This paper describes the implementation of a system that automatically learns selectional restrictions for individual senses of polysemous verbs from subject-object relationships. The selectional restrictions are inferred from an adaptation of decision tree induction, and are bound to the syntactic relations that realize them as part of a move toward automated construction of verb predicates.

Introduction

The primary aim of the present work is to learn selectional restrictions for use in the automated construction of verb predicates for individual meanings of polysemous verbs. Our learning algorithm for discovering selectional restrictions is grounded in the WordNet ontology for nouns (henceforth, WN), which provides hierarchical categorization for nouns in the English language (Miller 1998). The selectional restrictions for the arguments of a verb are bound to the grammatical relations that realize them, as inferred from decision tree induction. Here, we focus strictly on subject-object relationships and their ability to restrict verb meaning.

The largest collection of verb predicates to date is that of Gomez, which touts over 3000 predicates, mapping over 98% of the WN ontology for verbs (Gomez 2007). The predicates have been hand crafted over the course of a number of years – a process requiring expert familiarity with the WN ontology for nouns. One contribution of the present work is that the algorithm can learn selectional restrictions for the remaining verbs automatically, without any need of expert knowledge from the end user. The algorithm also exploits the gain mechanism of decision tree induction in order to avoid over- or under-specification of selectional restrictions, both of which would render verb predicates ineffective.

In the section that follows, we describe our learning algorithm, giving details of the implementation decisions we faced in this novel approach of adapting decision trees for the task of learning selectional restrictions.

Learning Algorithm

Our algorithm for learning selectional restrictions proceeds as follows. First, we select a target verb and extract its verb

senses from WN 2.1, along with example sentences and definitions given in its overview. We present this information to a user (trainer) of the system, and collect a small set of representative sentences for each verb sense.

The training sentences are then parsed and converted to a set of input data for decision tree induction. This input consists of the ontological categorization, also provided by WN, of the subject-object NPs under the scope of the verb, as well as a flag indicating the syntactic realization of these constituents. The result is a tree in which classification derives from an examination of these attributes. In the final stage of the learning algorithm, we apply a variety of refinement mechanisms to both eliminate decision points that cannot be used to derive selectional restrictions and precipitate selectional restrictions not indicated in the decision trees initially. Ultimately, we extract the ontological categories from the decision trees that form the basis for our verb predicates.

Training

When the user chooses a target verb to train on, they are presented with overview information from the WN verb ontology (Fellbaum 1998). This consists of glosses and example sentences for each sense of the target verb. The task facing the user then is two-fold: to instantiate predicates for the verb, and to give example sentences for each predicate.

The user must instantiate a new predicate by first giving it a label and then providing sentences to represent that predicate in training. The user has flexibility in choosing example sentences for training the system, although they are restricted to providing between one and four sentences per predicate. This restriction not only ameliorates the training task, but also prevents over- and under-training on individual predicates – a problem that would otherwise give the induction algorithm a skewed view of the verb’s usage and impede its ability to navigate the underlying noun ontology to discover selectional restrictions. The only other restriction placed on the trainer is that they should give at least one example sentence with and without an object in the case of ambitransitive verbs (such as “peel;” e.g., “Mary’s skin peeled” and “Mary peeled [after getting a bad sunburn]”).

Representation

The predicate labels given to example sentences in training serve as our classes in decision tree induction, and the sen-

tences themselves are the objects. To proceed, then, we need to enumerate attributes for our objects and define the values of those attributes for each training example.

The attributes derive from the subjects and objects seen in training. In the present work, we discard PPs, particles, and predicate complements, focusing instead on selectional restrictions for subject-object relationships only, and their ability to disambiguate verb meaning.

From each of these NPs under consideration, we first extract the head noun and query WN for its hypernym chain, consisting of all categories from the noun itself up to the *entity* concept, which serves as the root of the hierarchy. We append to these concepts the syntactic relations (henceforth, SRs) that realized the corresponding head nouns, and instantiate them as attributes in the system. The resulting attributes, all of the form $\langle concept \rangle _ (\langle SR \rangle)$, constitute the majority of the attributes. Each of these attributes may take on the values *true* or *false*, based on whether the concept categorizes the head noun of the corresponding argument in a sentence. There is also a $[null] _ (\langle SR \rangle)$ attribute for each SR, which is set to *true* if that SR is not realized in a sentence.

The final attribute for each training example is the class to which it belongs, which can take on the value of any of the predicate names for the given verb.

Finally, rather than having one decision tree per verb (in which the resulting classes correspond to the names of the predicates instantiated for each verb sense), we prefer to create a unique decision tree for each sense of a verb. We choose this approach, rather than having a single, unified DT, because the unified DTs tend to delineate into complex mazes of ontologically infeasible, nested decision points.

To achieve this system of DTs, we create a copy of the input data for each of the target verb's predicates. Then, for each predicate's input data, we replace any class names other than the predicate's name with *other*. The decision trees are created using the ID3 induction algorithm (Quinlan 1986). (An investigation with C4.5 proved fruitless; the algorithm was too tolerant to noise, which frequently resulted in the pruning of useful, even essential, selectional restrictions.) Each of these binary decision trees for a verb then undergoes refinement and post-processing to give rise to the selectional restrictions we use in our final predicates.

Extracting Selectional Restrictions from the DTs

At their deepest levels, the DTs contain decision points of the following types:

- (1) $[(\langle category \rangle _ (\langle SR \rangle)) (T \text{ predicate-name}) (F \text{ other})]$
- (2) $[(\langle category \rangle _ (\langle SR \rangle)) (T \text{ other}) (F \text{ predicate-name})]$

To build a verb predicate, though, we must automatically extract selectional restrictions from the verb sense's corresponding DT. To do so, we first go to the deepest level of the tree, where the arguments of the verb have been subjected to the most stringent restriction of their noun senses. There, we find rules of the types indicated above. We note that rules of type (2) do not actually indicate which nouns can be included under the given SR, and therefore do not represent viable selectional restrictions. Only rules of type (1) actually indicate selectional restrictions – concepts that, when categorizing an argument of the verb, indicate that the

sentence may fall under the predicate in question. In some cases, these concepts are tied with others in terms of gain. We deviate from a traditional approach to DT induction by listing all attributes that are tied. Then we admit all of those concepts belonging to rules of type (1) to our predicates as selectional restrictions. We do, however, apply one stipulation: if there are ten or more such concepts listed for a particular SR, then we discard all of them. Large lists of concepts tend to arise only when the DTs have encountered an ontological quagmire through which they cannot navigate, generally created by high level noun sense ambiguity among arguments seen in training.

Once we have extracted rules of type (1) and discarded rules of type (2) from the deepest point in our DT, we eliminate their corresponding attributes from our training data, and recreate the DT for the given predicate. This exclude-and-reiterate mechanism essentially forces the DT to perform its categorizations on other concepts within the ontology, giving rise to further selectional restrictions, often for other SRs, and often at different levels of specificity from the selectional restrictions we have already accumulated. We repeat this process until either we have run through five iterations of the algorithm, or the input becomes so barren that the induction algorithm cannot perform accurate categorization (thereby halting automatically).

In cases where we fail to derive selectional restrictions on a particular SR, we design the predicate to permit any head noun to be categorized by that SR. Finally, in order to avoid under-specification of arguments (and, therefore, over-inclusion by the predicates during semantic interpretation) we eliminate the concepts *entity*, *physical entity*, *abstract entity*, and *abstraction* from our selectional restrictions on a SR, as long as doing so leaves at least one concept intact.

Conclusions

Predicates produced for randomly selected verbs with this method have shown promising performance at semantic interpretation tasks, with precision and recall often in excess of 90%. A noted exception is with verbs relying on PP complements for disambiguation. Our next task is to add an isolated layer of DTs to our algorithm to derive selectional restrictions for PPs. Once this is completed, the predicates will be advanced enough to proceed with a fair comparison to a mature gold standard, such as Gomez's verb predicates, in tasks of semantic interpretation.

References

- Fellbaum, C. 1998. A semantic network of english verbs. In *WordNet: An Electronic Lexical Database*. The MIT Press. 69–104.
- Gomez, F. 2007. Automatic semantic annotation of texts. In *Proceedings of the GLDV-2007 Workshop on Lexical-Semantics and Ontological Resources*, 59–66.
- Miller, G. 1998. Nouns in wordnet. In *WordNet: An Electronic Lexical Database*. The MIT Press. 23–46.
- Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.