Summary of Research on Verb Algorithm Enhancements

Justin M. Del Vecchio

Summer 2003

Independent Study

Abstract

The first part of the summer research focused on reading papers, articles and selections from books that deal with contextual vocabulary acquisition and determining the meaning of unknown verbs from context. A written review was supplied for each of the selections read and eventually a bibliography of papers relating to the contextual vocabulary acquisition of unknown verbs was created. The second part of the research was applying the knowledge gained from the readings by proposing enhancements to Ehrlich's verb algorithm. The enhancements outline a method for integrating Levin's verb alternation categories into the existing verb algorithm so an unknown verb can be assigned to an alternation. The enhancements are written in pseudocode so future researchers working on the algorithm can implement them. I wrote two accompanying papers that further detail my summer research: "Review of Papers Read for the Verb Algorithm" and "Proposed Enhancements to Ehrlich's Verb Algorithm". Both are included in this paper.

Description of Contextual Vocabulary Acquisition and Ehrlich's Verb Algorithm

Consider a passage that contains a word unknown to the reader. When the reader is confronted with the word she might use the context it is presented in and her personal background knowledge to create a definition for the unknown word. In doing so the reader engages in contextual vocabulary acquisition (CVA) to generate a definition for the word that may not be an exact dictionary definition and is open to further refinement when future instances of the word are encountered. Sternberg (1987) showed that readers who were given explicit instructions to figure out the meaning of words from context did better at defining new words than those who used no process at all or relied on word memorization. An analytical process for defining words based on context has been modeled on a computational level using the Cognitive Agent of the SNePS System-an Intelligent Entity (Cassie) and the noun/verb algorithm created by Ehrlich (1995). The model calls for Cassie to mimic the mind of the reader. She has Semantic Network Processing System (SNePS) representations of the passage as well as background information that the reader would be expected to have and use to arrive at deductions. Ehrlich's noun/verb algorithm, a combination of LISP and the SNePS User Language (SNePSUL), is run on the SNePS representations and yields a definition for the unknown word that is accurate based on the input information.

The primary focus of the CVA research is to have Cassie's mind mimic the mind of a human when asked the meaning of a word from a piece of text she has just read. Her mind contains two related entities after reading a piece of text: First, the text passage itself and second, a ``toy" knowledge base that is a series of antecedent knowledge related to the text being read (Rapaport & Kibby, 2000: 7). Ehrlich's verb algorithm serves as Cassie's reasoning mechanism, allowing her to examine the contents of her mind with respect to the word that is to be defined. It is a set of steps that analyzes the passage and background knowledge representations, searching for clues in order to create a definition of a word from context. Using her knowledge base, representations of a text passage and Ehrlich's verb algorithm, Cassie can hypothesize the definition of a verb in a fashion similar to a human using a set of predefined steps. The definition created by the verb algorithm is not static, changing each time Cassie is updated with additional text passages or background knowledge that make reference to the verb to be defined.

My Role in the CVA Project

My role in the project is to research Ehrlich's verb algorithm, both defining its existing behavior and making recommendations with respect to possible enhancements. During the spring of 2002, I researched and documented the current implementation of Ehrlich's verb algorithm. During the summer of 2002, I made attempts at revising the verb algorithm by addressing issues I had discovered that spring. Written summaries for both efforts can be found at: http://www.cse.buffalo.edu/~rapaport/CVA/cvapapers.html. Summer research for 2003 began in early May when I met with Dr. Rapaport. At that time, he proposed I continue my work on the verb algorithm but with a different approach. There were to be two parts to my research, one, where I researched papers relating to CVA and another where I wrote pseudocode enhancements for the verb algorithm.

In part one, I was to accumulate as many papers, articles and book selections relating to the CVA of unknown verbs as possible. The listing would serve as a comprehensive bibliography on the topic. I would also write a summary for each, including in the summary suggestions for enhancements to the existing algorithm, proposals for entire new modules for the algorithm or comments on topics of general interest contained in the reading selection. I would begin by reading papers and journal articles dealing with Schank's theory of conceptual dependency. Conceptual dependency is of interest to the CVA of unknown verbs because the primitives provided by Schank mainly focused on actions (Lytinen 1992: 62).

In part two, I was to apply the knowledge I had gained from part one by proposing a set of enhancements to the verb algorithm. The enhancements would be written in pseudocode and not directly incorporated into to the verb algorithm. This would allow Dr. Rapaport to comment on the proposed changes as well as make it easier in the future for researchers either than myself to implement them.

What Was Accomplished

Over the course of the summer I read twelve selections dealing with topics either directly or indirectly related to the CVA of unknown verbs. A majority of the papers reviewed, ten, dealt with Schank's theory of conceptual dependency. Schank proposes between ten and twelve categories for his conceptual primitives. I was looking for a way to place an unknown verb into one of these categories based on the syntactic and semantic characteristics of the sentence in which the verb occurred. The benefit of using conceptual dependency primitives to groups verbs is that in order to be a good primitive, it must support a cluster of reliable inferences, making the implicit explicit (Lytinen 1992: 53). If a sentence can be read and its verb placed in a conceptual dependency category, then a number of pre-defined inferences can be made about the functionality of the verb and the purpose of the noun participants. This is not quite as good as having an exact definition of the verb but gives insight into the nature and meaning of the verb, especially with respect to the sentence it is used in.

The difficulty with this line of research is that Schank did not envision the conceptual dependency categories (ACT categories) to be used with verbs whose meaning is unknown. His theory of conceptual dependency is predicated on the fact that the verb's meaning in known beforehand and placement into a conceptual dependency category is a triviality. In our case, all that is known is the other components of the sentence and not the part binding them together, the verb. Placement into a conceptual dependency category becomes more difficult when the verb is unknown.

The only set of syntactic clues provided by Schank for placing sentences into ACT categories is his concept of *conceptual cases*. Schank points to the importance of conceptual cases, stating "We use conceptual cases as the basic predictive mechanism available to the conceptual processor" (Schank 1972: 568). Each conceptual dependency category is supported by an underlying combination of conceptual cases, of which there are four. For a sentence to fit a conceptual dependency category, it must have a specific conceptual case supporting it and chosen from a set of pre-defined conceptual cases. It is, admittedly, a confusing topic and a complete description of Schank's conceptual cases can be found in the paper I wrote titled "Review of Papers Read for the Verb Algorithm". The verb algorithm would need to recognize when a syntactic construct reflects an underlying conceptual case and which ACT categories allow the conceptual case. Conceptual cases are the only formal syntactic and semantic evidence Schank provided for placing verbs into ACT categories without having to know the meaning of the verb.

It became evident that Schank did not provide enough syntactic or semantic clues to create an algorithm that would examine a sentence and place it into one of the ACT categories. I wondered if there existed research that categorized verbs into similar groups based only upon the syntax of sentences using the verb. I then remember my research from the summer of 2002 and a book I had read by Levin (1993) about English verb alternations and their classes. I re-examined the book and discovered that of the 82 verb alternations proposed by Levin, a large number categorized groups of verbs based upon the prepositions commonly used for the object and indirect object. This meant that for a sentence with an unknown verb, an examination of the prepositions used for the object and indirect object could identify the verb as being a member of a group of verbs consisting of, say, 25 members. Therefore, the prepositions could be viewed as a unique combination that classified an unknown verb as possibly belonging to a specific group of verbs.

As an example, consider the following two sentences that use the prepositions *with* and *against* and the verb *whack*:

John whacked the stick against the fence.

John whacked the fence with the stick.

The prepositions *against* and *with* can both occur as the precursor to the indirect object for only a handful of verbs. In this case there are 24 verbs that use both *against* and *with* for the indirect object. Levin calls this set of verbs the Alternating Hit Verbs. Levin also relies on an examination of the semantic content of sentences, in addition to syntactic content, for many alternations. For instance, an alternation might be predicated on the fact that it uses the preposition *to* for the indirect object and the object is a *physical object* and the indirect object has *animate qualities*.

The goal of my research was to take all the alternations and turn them into an algorithm that would take a sentence containing an unknown verb and determine which, if any, verb alternation it belonged to. If the sentence with the unknown verb contained a combination of prepositions, or possibly semantic information, attributable to a specific verb alternation, it would be linked to the alternation. The list of verbs contained by the alternation might be similar to, or contain an identical match of, the one in question.

The pseudocode for the enhancements based on Levin's verb alternations generally work as follows:

If someone wants to define a verb

Then find all example sentences that use this verb and cycle through each. For each sentence:

> **If** the sentence uses a combination of prepositions and semantic content that matches the pattern of a particular verb alternation,
>
> **Then** assign the verb to that verb alternation and store the alternation

Return all the possible verb alternations this verb may belong to

The algorithm described above is deceptively simple and there are topics still open to debate. In my paper "Proposed Enhancements to Ehrlich's Verb Algorithm", I address the following unanswered topics: 1. How prepositions should be represented in SNePS networks and 2. How the semantic information searched for by the pseudocode algorithm should be represented. Any researcher working with these enhancements in the future will need to address both issues.

I probably reviewed 30 of the 82 verb alternations and was able to turn 22 of them into a component of the pseudocode algorithm. The remaining eight alternations that were not in the pseudocode were either too broad or too narrow to be of any use. The pseudocode code enhancements are 11 pages in length and were derived from 27 sample verbs from 11 of Schank's ACT categories.

Next Steps for the Immediate Future

When I began translating Levin's verb alternations into pseudocode, I had two possible ways of going about the task. The first would have been to take the alternations in sequential order and translate them into pseudocode. The second would have been to take a set of sample verbs and translate only the alternations covered by those verbs into pseudocode. I choose the latter, using 27 sample verbs from 11 of Schank's ACT categories. There are a couple reasons this turned out to be a wise choice. First, I noticed that the verbs selected spanned across all Levin's verb alternations, providing me with a comprehensive exposure to the verb alternations I doubt I would have gained had I translated them sequentially. Second, taking a sampling of many verbs allowed me to see that some verbs from different ACT categories shared almost identical verb alternations. This phenomenon will bear further investigation.

Future researchers will have to translate the remaining verb alternations into pseudocode. My advice would be to first review the alternations I have covered so far to get a feel for what to expect. After doing that, the researcher can choose to finish off the rest of the alternations by doing them either sequentially or by choosing another set of sample verbs and seeing how many verb alternations are covered. It might be wise to create another list of sample verbs based off Schank's ACT categories and proceed in that manner until all the verb alternations have been covered. A comprehensive list of the verb alternations that I have covered so far is in my paper "Proposed Enhancements to Ehrlich's Verb Algorithm".

Another topic of research is to determine is if it would be possible to refine a listing of verbs returned by a particular verb alternation based on the sentence the verb is used in. For example, the *Apart* Reciprocal Alternation - Alternating Verbs returns the following list of verbs:

- Split Verbs: blow, break, cut, draw, hack, hew, kick, knock, pry, pull, push, rip, roll, saw, shove, slip, spit, tear, tug, yank

Would it be possible to return an even smaller subset of these verbs? Dr. Rapaport proposes that some sort of test be created to take the verbs and further divide them into separate ACT categories. I have given no consideration to what these tests might look like or how this mechanism would work. The goal is to be able to say something like "I know this is a PTRANS verb" as opposed to "I know this verb is either blow or break or cut or draw or hack or hew…"

Another immediate goal is to find a series of sample sentences to run the verb algorithm against. The sentences could be either made-up or from an actual source. I knew that one of the verbs for which representations have been created is *pry*, so I made certain to examine the verb alternations of pry and include them in the pseudocode enhancements. What I have not done is look at the representations created by Chris Becker in his paper "Contextual Vocabulary Acquisition: On a Representation of Everything but the Word `Pry'" located at http://www.cse.buffalo.edu/~rapaport/CVA/Pry/cva_spring2003.html. It would be worthwhile to read this paper and see if the representations proposed by Chris would yield any results if run against the proposed verb enhancements. With respect to finding sample sentences, I would concentrate my efforts on the verb *pry* and try and find additional sentences, some of which can be found in Chris' paper. Another approach might be to look at the alternations covered in the enhancements and find a few verb groups that return particularly hard verbs like *purloin*. Then, based on these difficult verbs, find example sentences that use them and verify what the pseudocode enhancements would return for each.

Next Steps for the Far Future

First, what case frame strategy do future researchers choose for incorporating prepositions into SNePS representations? My paper "Proposed Enhancement's to Ehrlich's Verb Algorithm" covers three different approaches for adding prepositions into representations, listing pros and cons for each. The over-arching question to be resolved is, 'If SNePS representations are to be universal in nature, is it appropriate to include information about prepositions that are language specific, in our case specific to English, in them?' If future researchers choose to include information about prepositions directly in the SNePS representations, then work on merging the verb alternation enhancements into the verb algorithm can begin immediately. If future researchers choose to include information about prepositions in the grammar, then the grammar will need to be written first and the verb algorithm enhancements later. Which approach is better is still a topic of debate.

The representation of semantic information needs to be addressed as some verb alternations rely on finding the existence of certain semantic representations in order to categorize a verb. For instance, objects that are *locations* or have *animate qualities* are referenced in numerous verb alternations and are actively searched for in the verb algorithm enhancements. This information needs to be represented in a systematic and predictable manner. Whoever is responsible for creating SNePS representations from text, an automated parser or knowledge engineer, needs to be aware that certain properties or characteristics should be represented in a specific manner so they are found by the verb algorithm enhancements. I have kept a list of all the semantic information actively searched for by the verb algorithm in my paper "Proposed Enhancements to Ehrlich's Verb Algorithm". Any time a new alternation is added to the verb algorithm and semantic content is part of its definition, this list will need to be consulted and possibly updated.

The third major task that needs to be done is to merge the verb algorithm enhancements with the existing verb algorithm. The functioning of the existing verb algorithm is summarized in my paper "A Computational Definition of 'Proliferate'" at http://www.cse.buffalo.edu/~rapaport/CVA/cvapapers.html. Future researchers should read the description located here for how the current verb algorithm functions and then analyze the algorithm's Lisp code for a thorough understanding. I do not believe it will be an overly difficult task to merge the two algorithms and the majority of the time will be spent on deciphering Lisp as well as manipulating it to get the desired results.

Conclusion

The summer research for 2003 has been fruitful. I was able to examine Schank's theory of conceptual dependency and determine it is of value with respect to providing a definition for an unknown verb. If a verb can be placed into specific ACT categories, then a host of inferences can be made about the verb and the noun participants of the sentence. The difficulty with this approach is the lack of syntactic or semantic clues that would allow an unknown verb to be placed into a conceptual dependency category. I examined Levin's verb alternations to determine if these could provide the syntactic and semantic clues necessary for categorizing an unknown verb. This research yielded positive results and it quickly became evident that by examining the syntactic and semantic make-up of a sentence, the likelihood was good the verb in the sentence could be placed in one of Levin's verb alternations. The pseudocode details how this can be done algorithmically. The next step is to determine if the set of verbs returned by an alternation can be further refined and categorized into Schank's ACT categories.

References

1. Ehrlich, Karen A. (1995), *Automatic Vocabulary Expansion Through Narrative Context* (Buffalo: Department of Computer Science and Engineering of State University of New York): 74-80 and 96-108.
2. Levin, Beth (1993), *English Verb Classes and Alternations* (Chicago: The University of Chicago Press).
3. Lytinen, Steven L. (1992), "Conceptual Dependency and its Descendants", in Fritz Lehman (ed.), *Semantic Networks in Artificial Intelligence* (Oxford: Pergamon Press): 51-73.
4. Schank, Roger (1972), "Conceptual Dependency: A Theory of Natural Language Understanding", *Cognitive Psychology* (3)4: 552-631
5. Sternberg, Robert J. (1987), "Most Vocabulary is Learned from Context," in Margaret G. McKeown & Mary E. Curtis (eds.), *The Nature of Vocabulary Acquisition* (Hillsdale, NJ: Lawrence Erlbaum Associates): 89-105.
6. Rapaport, William J., & Kibby, Michael W. (2000), "Contextual Vocabulary Acquisition: Development of a Computational Theory and Educational Curriculum", NSF grant proposal.

Review of Papers Read for the Verb Algorithm

Justin M. Del Vecchio

Summer 2003

Independent Study

Abstract

A goal of the summer's research was to read numerous papers dealing with topics relevant to the verb algorithm. This paper is a summary of the readings in chronological order. Initially, it was believed that the elements of Schank's theory of conceptual dependency would be of interest and the first articles reviewed deal with this topic. A progression is evident in the reviews as it becomes obvious that assigning an unknown verb to an ACT category requires an analysis of the sentence's syntactic makeup and I search for papers that directly address this topic. This leads to reviews of Levin's work (1993) on verb alternations and is the most fruitful research of the summer. Levin essentially categorizes verbs into like groups based on the syntax of the sentence they occur in.

Overview of Lytinen's Article "Conceptual Dependency and Its Descendants"

Conceptual dependency is of interest to contextual vocabulary acquisition of unknown verbs because the primitives provided by Schank mainly focused on actions (Lytinen 1992: 62). Verbs are commonly considered to be actions or the relation that binds the actors of sentences together. Schank provides 10-12 primitives, depending on the paper, which are base actions that all other actions can be mapped to.

Lytinen has reservations about the small number of primitives, stating "No one seems to take seriously the notion that such a small set of representational elements covers the conceptual base that underlies all languages" (Lytinen 1992: 51). Regardless of the correctness of Lytinen's claim, conceptual dependency spawned many research projects that could provide guidance for future versions of the verb algorithm. Taken alone, conceptual dependency provides an interesting strategy for guessing the meaning of verbs. Conceptual dependency has two core assumptions:

1. If two sentences have the same meaning, they should be represented the same regardless of the particular words used.
2. Information implicitly stated in the sentence should be represented explicitly. Any information that can be inferred from what is explicitly stated should be included (Lytinen 1992: 52).

From the above two points, we can infer that the following sentences have identical meaning:

1. "Tom walked from the park to the store."
2. "Tom jogged from the park to the store."

Here, the primitive would be a PTRANS representing the verbs 'walk' and 'jog' and the slots for PTRANS would be as follows

1. ACTOR: a HUMAN (or animate object) that initiates the PTRANS
2. OBJECT: a PHYSICAL OBJECT that is PTRANSed (moved)
3. FROM: a LOCATION, at which the PTRANS begins
4. TO: a LOCATION, at which the PTRANS ends.

What is interesting are the pre-defined slots provided for a PTRANS. Here, there is an 'actor', an 'object' and the 'to' and 'from' slots which pertain to the syntactic structure of the sentence. I would propose the following strategy for the verb algorithm. Consider, if instead of 'walk' or 'jog', the sentence were the following:

Tom ambulated from the park to the store.

The verb algorithm would look at the sentence stored in CASSIE, where the case frame used would include all pertinent information, namely the prepositions 'from' and 'to'. Based on the syntactic structure of the sentence, the verb algorithm could formulate that this verb has the primitive form of a PTRANS. How would it do this? It would take the representation and attempt to place the actors of the sentence into the predefined slots for the 10-12 different primitive types. In the example above, it would determine that since the sentence has the prepositions 'to' and 'from', and the nouns that would fill these slots are places, 'ambulated' is a PTRANS. This is, admittedly, a thumbnail sketch and has problems like "What if the representation used the prepositions 'to' and 'from' but was not a PTRANS?" Would it be possible for such a situation to arise?

One huge benefit of using conceptual dependency primitive is that in order to be a good primitive, it must support a cluster of reliable inferences, making the implicit explicit (Lytinen 1992: 53). For instance, in the above sentence the verb algorithm could determine it is dealing with a PTRANS and then apply a set of well-defined, as well as pre-defined, inference rules like "The OBJECT which was PTRANSed was initially in the FROM location and afterward was in the TO location." This works because conceptual dependency is founded on the idea that there is one inference rule for several representations that boil to the same canonical representation (Lytinen 1992: 52).

I think I should read one of Schank's papers on conceptual dependency, possibly from the 1972 book in the bibliography which Lytinen implies is better than the original article. What I would like to do is determine the set of primitives that the verb algorithm can use and for each member of the set, find if pre-defined syntactic structures exist. By this I mean do what was done for the PTRANS example above where the existence of the prepositions 'to' and 'from' hinted at the fact the verb was a PTRANS. I hope that patterns will exist for each of the primitives and that I can define these syntactic structures.

Another possible benefit of using conceptual dependence's primitive is the possibility for deep inferencing. Lytinen states "Representations could be become arbitrarily complex, sometimes making explicit a whole series of inferences that could be made" (Lytinen 1992: 53). I am not entirely sure what this would lead to but I am assuming it could help in defining the meaning of the word 'crying' in the following sentence:

John cried because Mary said she loved Bill.

Lytinen goes into a brief discussion of causal connections and causal chains that are probably worth future investigation. I am hoping the Schank book will shed more light on this.

The core of conceptual dependency primitives is that they represent the meaning of a text in canonical form to facilitate inferencing. If we can accurately guess which primitive category an unknown verbs falls into, based on the position of the actors, prepositions used, etc., we are immediately given a host of inference rules that can be applied to the sentence.

The drawback is that primitives try to find a canonical form. 'run' and 'walk' will both fall in the realm of PTRANS but how will the verb algorithm know the difference between them; running is quick movement while walking is relatively slow.

MEMORY Overview

MEMORY is program that makes inferences from text. For example:

John hit Mary. Mary's mother took Mary to the hospital. Mary's mother called John's mother. John's mother spanked John.

MEMORY would make inferences about the story like John's mother spanked John because she was mad at him for what he had done to Mary. What made MEMORY unique was that when it read a sentence, its inference rules were automatically applied (Lytinen 1992: 56). It is not clear if MEMORY would have any direct applications to a verb algorithm. I mention it because it makes automatic inferences, much like human's make uncontrollable inferences when reading, listening, watching, etc. It may be of use to the verb algorithm but only at a very high, conceptual level.

Plans, Goals and TOPS Overview

Lytinen provides sections on the notion of high-level, goal-based representations. The discussion on TOPs, thematic structures like the adage "closing the barn door after the horse is gone" which can have many different variables but maintains a common, thematic form, may have application to the verb algorithm. This is very high-level stuff and may be more interesting than actually helpful with respect to the verb algorithm.

Request-Based Parsing

This is a topic that may be of interest. Lytinen discusses Riesbeck's analyzer (Riesbeck 1972), which is a parser coupled with an associated lexicon. Requests were made of the parser and if the associated word definition existed in the lexicon, its meaning was sent back. For example, the dictionary entry of the verb "ate" contained a request to build an instantiation of the concept INGEST (the conceptual dependency primitive underlying eating and drinking) and request for a noun group after the verb which referred to a food item, which, if found, was placed in the OBJECT slot of INGEST (Lytinen 1992: 66). What is interesting about the process is that the parser also handled words that did not exist in the lexicon. Lytinen does not go into a great amount of detail about how this was done but I think it might be worthwhile to review this paper and determine how the parser handled unknown words. There are several examples of parsers that use request-based knowledge including Integrated Partial Parser, the Word Expert Parser and BORIS. These might be of interest as well.

Here is another possible approach for the verb algorithm. Riesbeck's parser worked as followed: The verb "ate" had a request for a noun group to its left, which was an ANIMATE. If such a noun group was found, it was placed in the ACTOR slot of INGEST. The verb algorithm might try the inverse of this. It would find an unknown verb. It would then look to the left and right and find the noun groups. Based on the type of noun groups and other know verbs it would do trial and error to see if an identical match existed that accepted the same type of noun groups. Of course, this would require a formal set of noun groups and supporting case frames.

MOPTRANS

MOPTRANS was a program that translated short newspaper stories about terrorism and crime. It focused on using syntax and semantics hand-in-hand. Semantics proposed various attachments, and syntax acted like a filter, choosing which of these attachments made sense according to the syntax of the input (Lytinen 1992: 68). This seems similar to how we would the verb algorithm to behave. For example, the sentence:

John gave Mary a book.

would have been represented as an ATRANS sentence in MOPTRANS, as 'gave' would have already been known. What is of interest is the way MOPTRANS assigned the ACTOR and OBJECT slots of 'gave'. The semantic analyzer would try to combine these representations in whatever way it could, concluding that the PERSON, John, could be either the ACTOR or the RECIPENT of the ATRANS. The parser utilized syntactic rules to properly assign the ACTOR and RECIPIENT, having an accompanying Subject Rule that would assign the PERSON to be the ACTOR of an ATRANS. The process and rules MOPTRANS used to come to its conclusions may be useful to the verb algorithm. It appears that it examined the syntactic structure and, based on a set of pre-defined rules related to syntactic structure, assigned the nouns to appropriate slots. This process is similar to how I envision the verb algorithm working; examining the syntactic structure to make inferences and assignments.

Overview of Roger Schank's Article "Conceptual Dependency: A Theory of Natural Language Understanding"

The underlying premise of Schank's theory is that there exists a conceptual base onto which all linguistic structures in a given language map (Schank 1972: 554). Conceptual dependency extracts the concepts that the words denote and relates them in some manner with those concepts denoted by the other words in a given utterance. It is quite possible for a sentence to realize numerous conceptualizations at the same time (Schank 1972: 556). For the purposes of the verb algorithm, the unknown verb will have a conceptual relation to the other words in the cotext and surrounding sentences that the algorithm attempts to fill in. It is probable that the concept represented by an unknown verb can be inferred by analysis of the syntactic and semantic makeup of the cotext and surrounding sentences of several examples passages using the unknown verb. This will be discussed in detail later.

Schank describes three elemental kinds of concepts that serve as building blocks for his representations (conceptual diagrams, called c-diagrams for short):

1. **Nominal** – Nouns. Called PPs or picture producer.
2. **Action –** Verbs. Called ACTs or base actions.
3. **Modifier** – Adjectives and adverbs. Called PAs for picture aiders (adjectives) and AAs for action aiders (adverbs).

Each of the conceptual categories relates to one another in specified ways (Schank 1972: 557) and Schank goes into detail about the 'dependency' aspect of conceptual dependency where two concepts in a relation are constrained by a governing concept and a dependent concept. In general, governors are ACTs and PPs while dependents are PAs and AAs, though this need not always be the case. What is of interest is the fact that ACTs, verbs, are usually the governors in a conceptual dependency relation and therefore central to understanding the relation. An example:

John hit his little dog

In this sentence, John' and 'hit' both act as governors with a two-way dependency existing between them. Both can be understood independently but it is the coupling of the two that must be present in order to form a conceptualization (Schank 1972: 558). For the purposes of the verb algorithm, I think this type of information is already captured in existing case frames as the governors relate to the subject and action of the sentence.

Analyzing a Sentence Using Conceptual Dependency

Schank describes a computational system for implementing conceptual dependency including the different components that would be necessary in order analyze a sentence. Below is a listing of the components:

1. Syntactic Processor
2. Conceptual Processor
3. Conceptualization-Memory Interface

4. Intermediate Memory
5. Long-Term Memory

So far, I have a good grasp on the first and second points. These would seem to have the most impact on future verb algorithms. The remainder of this paper will go into detail on points one and two, analyzing how they might apply to a verb algorithm for CVA.

Syntactic Processor

Schank states "It is necessary to be aware of the syntactic relationships that exist between words" and "The main functioning of the syntactic processor is to find the main noun and verb, secondary is the establishment of syntactic dependencies between them." (Schank 1972: 560). He proposes the following verb types (Schank 1972: 563):

1. vt – transitive verb - "The big man took the book."
2. vi – intransitive verb
3. vx – double object verb - "I gave the girl candy."
4. vio – indirect object verbs
5. vs – state verbs "I wish he was here."

This is similar to the functioning of the current verb algorithm; categorize verbs based upon transitive, intransitive, reflexive, etc. Schank requires this categorization for a predefined verb-ACT dictionary in which entries for known verbs would be stored and each entry further broken down based upon the verb type (transitive, intransitive, etc.)

An example of an entry in the verb-ACT dictionary would be:

take

Verb-type	***Conceptual meaning***	***Semantic category of X***	***Semantic category of Y***	***Semantic category of Z***
vt	X take Y to the	Human	phyobj	

Verb-type	***Conceptual meaning***	***Semantic category of X***	***Semantic category of Y***	***Semantic category of Z***
	possession of X			
vx	X take Y to the possession of Y	Human	human	phyobj
vt	X ingest Y to the innards of X	Human	medication	

Some observations based on the above table with respect to the verb algorithm:

1. **A preexisting dictionary of verbs must exist that the verb algorithm can refer to.** I think this will be necessary for the purposes of comparing unknown verbs with existing ones for pattern matching. If an unknown verb were to match a known verb based upon verb-type and semantic categories of the participant nouns, it might be the case that their meaning is closely related or possibly synonymous. This will require some work as example verbs for each of the ACT categories would have to be agreed upon and input into SNePS. However, I think it mimics the way a human would go about the problem. S/He would come across an unknown verb in a number of settings and apply known verbs to see first if it is grammatically legal and second if it makes sense.
2. **The categories of the participating nouns are important.** It appears that the noun participants must belong to some class and a standard for assigning nouns to classes must exists and be applied uniformly to sentences translated into SNePS. This is similar to what Dr. Rapaport had hoped I would come up with while researching Levine in the summer of 2002. In the first few versions of the verb algorithm I do not think this will be a huge issue as I foresee the syntactic analysis and breakdown of sentences taking a majority of the work. In later versions, this will become an important point.

Both points above are overarching concerns and stray somewhat from what I had originally hoped to find in reading this paper; a set of case frames that would allow for unknown verbs to be placed in one of the

10-12 ACT categories based on syntactic analysis. I will address this point later as Schank makes a host of different points that need to be discussed before I can make a prediction on this could be done.

The Conceptual Processor

The responsibility of the conceptual processor is enumerating the underlying relationships that readers of a sentence of text know exist based upon their background knowledge, the cotext and context of the sentence. Schank states, "The conceptual processor relies on the predictions that it can make from what it knows about the input it has already received in order to know how to deal with new input" (Schank 1972: 563).

Schank gives 12 rules that formally define the dependency relations between given categories of concepts. The rules depict the legal representations of concepts in c-diagrams but are a bit much for mention here. What are more important are the identical c-diagrams that arise from different sentences using different verbs but putting forth the same concept. Consider the following two sentences:

The man took a book.

I gave the man a book.

On the surface they appear to be different sentences representing different ideas. In one, the man takes a book in the other I give the book to a man. Conceptually, however, the same underlying action has occurred leading to identical conceptual dependency representations for both (Schank 1972: 567). 'Give' is like 'take' because both require a recipient and an original possessor of a specific object. The actual conceptual dependency relationship drawings have been left out because it difficult to recreate them using a word processor; they contain numerous symbols and flow both left to right as well as up and down. In both representations the verbs 'give' and 'take' are replaced with the generic action 'trans', for transfer, and there are 'to' and 'from' components for the original possessor and recipient.

Schank defines give and take as follows (Schank 1972: 568):

'Give' – a 'trans' where the actor and originator are identical

'Take' – a 'trans' where actor and recipient are identical.

He lists the following as 'trans' verbs: give, take, steal, sell, own, bring, catch and want. Any sentence that contains these verbs can be rewritten into conceptualizations with common elements that allows for recognition of similarity of utterances (Schank 1972: 568). I believe this has a direct impact on the verb algorithm as follows:

1. SNePS representations for known verbs could be made incorporating into the case frame information like the ACT category (PTRANS, MTRANS, etc.) the verbs belonged to.
2. Unknown verbs would also be placed in these case frames, filling out as much information as possible with the exception of ACT category membership.
3. Based on the similarity of case frames between known verbs and unknown verbs, deductions could be made as to what subset of existing, known verbs might the unknown verb be synonym for or closely related to.

Again, this requires the verb algorithm have a knowledge base of existing verbs that most people could be expected to know. I think it would be difficult for a person to understand an entirely new verb without having a set of known verbs off which to make their inferences from (Genter makes this point, though I will have to find the paper).

Schank states that if two sentences have the same meaning, one conceptual diagram should suffice to represent them (Schank 1972: 568). I wonder if the same could be said for case frames? I have not really thought about this in any great detail.

ACT Conceptual Cases

Dependents that are required by ACTs are called conceptual cases (Schank 1972: 568). Conceptual cases will be vitally important to a new verb algorithm based on conceptual dependency because they will serve as the filters that allow unknown verbs to be placed in ACT categories (PTRANS, MTRANS, etc.) Schank points to the importance of conceptual cases, stating “We use conceptual cases as the basic predictive mechanism available to the conceptual processor” (Schank 1972: 568). The four conceptual cases are:

1. Objective - “I ate an apple.”
2. Recipient - “I sent mother money.”
3. Directive - “He went home.”
4. Instrumental - “I ate the pudding with a spoon.”

ACT categories are particularly powerful because once a verb is realized to be in a specific category, predictions can be made thereby changing the basic analysis process (Schank 1972: 577). Verbs are placed into ACT categories based on which conceptual cases can be applied to them. There are only four conceptual cases and for any given ACT category, a certain set number of cases are required (Schank 1972: 574). This number can be as few as none or as many as three.

Syntactically, the four conceptual cases can be realized in English by the following constructions following the verb where N is the PP for that case, or in the instance of instrument, N is the ACT of the instrumental conceptualization if it can be so realized or else it is the objective PP (Schank 1972: 574):

Objective	(nil) N
Recipient	(to, from) N
Instrumental	With N, b N(-ing), using N
Directive	(to, from, toward) N

The table above is the first set of syntactic clues provided by Schank for placing sentences into the requisite conceptual case. Hopefully, this list can be expanded or clarified by further readings.

What will be key for the verb algorithm is realizing when a syntactic construct reflects an underlying conceptual case and which ACT categories allow the conceptual case. Schank had the benefit of a predefined ACT-dictionary in which to find a verb's ACT category. In our case, we will have to make a deduction as to which ACT category the verb falls into based on the syntactic makeup of the sentence as well as its surrounding context.

A Possible Verb Algorithm Outline Based on Most Recent Schank Findings

It is apparent that analyzing a single sentence is not going to allow for placement in an ACT category. The process will be repetitive and go as follows:

1. Read a new sentence
2. If possible, assign a conceptual case to the sentence. If lucky, the syntactic makeup of the sentence will denote it as a single conceptual case.
3. Add this conceptual case to the unknown verb's definition
4. Evaluate which ACT categories require this conceptual case along with any other conceptual cases the unknown verb has been assigned
5. Remove ACT categories that do not apply to the unknown verb
6. If one ACT category it is a good sign.

Other steps will need be involved here. For instance, I envision that sometimes we will apply two conceptual cases to a sentence (though only one can be the correct) in an effort to whittle down the ACT categories. Hunches like these will need to be monitored and revisited to determine if they are still applicable.

Overview of Schank and Rieger's "Inference and the Computer Understanding of Natural Language"

The primary focus of this paper is how to take an ACT primitive and to it apply a set of predefined inferences. Each set of predefined inferences applies to a specific ACT and not a specific verb. For example, the set of inferences that applies to "John sold a book to Ellen" will be different from the set of inferences for "John sold Ellen on his idea." despite the fact both sentences use the same verb. The paper introduces some different categories of inferences that can be applied to ACTs but, unfortunately, does not provide a comprehensive listing of the inferences that can be applied to each ACT category. Instead it provides a few inference rules and concentrates on how the rules would apply to conceptualizations. The paper is not clear on whether a comprehensive list of the inferences for each ACT category exists. It may be an assumption that such a list exists but it would be up to the implementer of a conceptual dependency system to create the inferences rules for each ACT primitive.

The paper is radically different with respect to the ACT categories from Schank's 1972 introductory paper on conceptual dependency. This is the first paper to put forth the list of twelve ACT categories researchers are commonly familiar today. The major ACT categories like ATRANS, PTRANS, MTRANS, PROPEL, etc. are defined in this paper and have remained consistent to the current time. However, there always seems to be a fringe element of ACT categories that Schank leaves open to change or reorganization like PLAN or MBUILD.

The paper does not provide a comprehensive listing of the verbs that fall in each ACT category or the syntactic structure of each ACT category. I had hoped the paper would cover those two points in greater detail but it does discuss numerous points that would apply to the verb algorithm. I have expanded on these points.

Inference versus Deduction: The Verb Algorithm Will User Both

The paper goes into a lengthy discussion of the difference between inferences and deductions. This applies to the verb algorithm because it would make a mixture of both inferences and deductions with, I believe, a larger percentage of inferences being made than deductions on the whole.

An inference is a new piece of information generated from another piece of information and which may or may not be true (Schank 1974: 120). Inferences differ from deductions in the following way: An inference is an implicit motivation to generate new information from old while deductions are performed only upon demand, as would be the case in a theorem-prover (Schank 1974: 120). The paper points out that an inference is like circumstantial evidence. I imagine that a verb algorithm would not always function in the realm of deductions and would continually be making inferences or, more plainly, following it hunches base upon sound reasons. The following are five features of inferences that depict how inferences differ from deductions:

1. An inference is an implicit motivation to generate new information from old.

2. An inference need not be necessarily true.
3. A system using inferences must maintain information as to whether the inferences it makes may or may not be true.
4. Inferencing, unlike deductions, does not start out with a goal in mind.
5. A memory dealing with an inference needs a manner of belief revision

Point three will be very important for the verb algorithm. It will need to keep an index of what it believes to be true and some a weighted system of which of these beliefs has a higher likelihood of being true than the others. The verb algorithm will have to make use of belief revision.

Inferencing and Parsing and a Possible Implication for the Noun Algorithm

A meaning representation contains "each and every concept and conceptualization that is explicitly or implicitly inferred to by a sentence" (Schank 1974: 121). If the sentence were:

John bought a book.

A parser would need to realize that following implicit and explicit information exists:

Explicit	***Implicit***
John	Two actions of transfer exist
Book	Money was exchanged

The explicit information is directly available from the sentence while the implicit information is not. The paper states that it is the job of the parser to extract this implicit and explicit information. While it is easy for the parser to pick out explicit information, another mechanism is needed to do inferencing, the adding on of probably correct information (Schank 1974: 121). This mechanism to do inferencing is called the "conceptual analyzer".

It would seem that a conceptual analyzer would be of great value to both the noun algorithm and verb algorithm. The verb algorithm desires to find the ACT category a verb falls into and then apply inference rules to the sentence. The noun algorithm, however, would already know the category of the ACT and could make numerous inferences from the sentence for unknown nouns.

Syntactic Rules of Conceptual Dependency Represented as SNePS Case Frames

There are sixteen conceptual dependency syntax rules. According to the Rich book, there are fourteen rules. One task **to do** is to map the sixteen categories to current case frames. This seems like it will go pretty easily as most of the rules can be reinterpreted as current SNePS case frames with a few minor adjustments.

The Primitives

It is the job of the primitives to link similar information so that inference rules need not be written for each and every individual verb, rather written for the ACTs that contain numerous verbs of the same conceptual fabric. The following eleven primitives seem to be stable representation of the ACTS:

1. ATRANS - Transfer of an abstract relation such as possession, ownership, or control
2. PTRANS - Transfer of physical location of an object
3. PROPEL – The application of a physical force to an object
4. MOVE – The movement of a body part of an animal
5. GRASP – The grasping of an object by an actor
6. EXPEL – The expulsion from the body of an animal into the world
7. MTRANS – The transfer of mental information between animals or within an animal.
8. CONC – The conceptualizing or thinking about an idea by an animal.
9. MBUILD – The construction by an animal of new information from old information
10. ATTEND - The action of directing a sense organ towards an object.
11. SPEAK – The action of producing sounds from the mouth.

Each ACT has one of four conceptual cases: OBJECTIVE, RECIPIENT, DIRECTIVE or INSTRUMENTAL. Each act takes two or three of these cases obligatorily or none optionally (Schank 1974: 124). The numbers for how many conceptual cases apply to an ACT differ from Schank's initial 1972 paper. In addition, the INSTRUMENTAL conceptual case is a complete conceptualization in itself.

Applying Predefined Inferences to ACTs

Pages 130-139 detail a program that makes inferences based upon a conceptualization. This is too advanced for current verb algorithm discussions as we are still addressing the problem of how to place an unknown verb into its appropriate category. This section will be of use in the future.

Possible Points to Incorporate in the Verb Algorithm

1. Based on the examination of the sixteen syntax rules, determine any new arcs that need to be added to existing case frames.
2. Only animate objects may serve as actors for the ACT categories except for the PROPEL case. Implication: the slot for the object in the object/act case frame will always need to have the property attached to it that defines whether it is animate or inanimate. This is one check that can be applied to an unknown verb to determine its category. I would like to make a hierarchy of these checks and have logic behind them but for now I can only make a collection of all the prospective checks I would like to make.
3. Instrumental inferences can always be made although the degree of accuracy differs depending on the particular acts (Schank 1974: 127). Implication: no matter what the ACT, an instrumental inference can be made.

Overview of Conceptual Dependency Section in "Encyclopedia of Artificial Intelligence (EAI)"

The entry in the book is not very long and has only five pages dedicated specifically to Schank's theory of conceptual dependency. The section contains a few topics not covered in other texts already reviewed and I will review these areas below as they apply to the verb algorithm. There exists an interesting section on conceptual dependency and natural-language understanding (pgs. 669 - 670) that offers a simplified notation system for Schank's complex notation system. The notation offered in the book is similar to the arcs used by SNePS and translates conceptual diagrams to frames.

Conceptual Dependency Slots and Rules

A conceptualization has semantic constraints on the kinds of entities that can fill its slots (EAI 1987: 195). Some of the conceptual dependency rules (there are fourteen rules for conceptual dependency) can be applied to any type of object that fills the slots of an action. Therefore, a rule that would apply to any type of slot filler and could not act as a constraint condition in the verb algorithm. Some of the conceptual dependency rules are very specific as to the action and the values that fill the slots based on the action's type. This implies that some ACTs are constrained by the slot fillers. For example, MTRANS is the transfer of mental information between animals or within animals. If the actor of a sentence containing an unknown verb is an inanimate object, say an automobile, the likelihood of the verb being an MTRANS is remote. Unfortunately, a comprehensive listing of such inferences or constraints is not given. Either such a list does not exist or such a list is unimportant when approaching conceptual dependency from the point of view that the verbs ACT type will always be known beforehand.

It is apparent that there are two approaches to guessing which type of ACT an unknown verb belongs to:

1. Infer meaning based upon the membership or properties of the slot fillers.
2. Infer meaning based upon the syntactic makeup of the target sentence. This was our original goal for the summer research.

For both points above, I would propose looking at Levin's verb categories with two purposes in mind, the first more in line with the goal of the research and the second a nice addition:

1. Map Levin's verb categories to the ACTs. Based on the mapping, determine the typical syntax of the sentences and create case frames based off those. Levin's book is one of the few places where a comprehensive list of categorized verbs exists as well as examples of usage exits. It is one of the only avenues where I can hope to find case frames for the ACT categories.
2. Examine the Levin categories and determine the constraints on the slot fillers. For example, which categories of verbs always require animate objects to be the actors, which categories always need a location to be the indirect object, etc.

How Verbs Fall into ACT Categories

It is stated that the ACTS are not category names for verbs but should be considered elements of those verbs (EAI 1987: 196). I believe no comprehensive list of verbs for each ACT category exists because verbs can span multiple categories based upon their syntactic usage. It would not make sense to have a listing of the verbs that fall into each category. Instead, a listing would consist of example sentences of when a verb qualifies for one particular category as opposed to another. For example, consider the two sentences that use the verb 'give':

John gave the book to Mary.

John gave me a good idea.

In the first sentence, 'give' is used as PTRANS, in the second as MTRANS. It would seem that the constraints an algorithm could use to determine verb categorization would be a combination of the class membership or properties of the slot fillers as well as the overall syntax of the sentence. This will be a very involved task.

Translating a Sentence into a SNePS Representation Based on Schank's Schema

A big task will be working conceptual dependency representation into SNePS representations that maintain all pertinent information. Consider the following sentence:

John gave a ball to Mary.

This would require the following conceptual dependency representation:

ATRANS	
Rel	Possession
Actor	John
Object	Ball
Source	John
Recipient	Mary

If the verb were not known to be an ATRANS, how would an algorithm assign a source or a recipient? It would not know to. I imagine that the algorithm would need to figure out all potential ACTs the unknown verb could be, fill the slots for each type and maintain multiple, possible case frame representations for an unknown verb. The algorithm would need to use future examples to cull out the ACTs that are not viable, continually updating the possible case frames for unknown verbs. It would be a form of belief revision specific to the verb algorithm.

Overview of 'The Primitive ACTs of Conceptual Dependency', Memo to Yale University

This is a very informative, though brief, description by Schank of the primitive ACTs. It gives an overview of how ACTs function in conceptual dependency and then describes the ACTs in greater detail than any book or article I have read to this point. This memo is very informative and lays out how ACTs serve as the building blocks of any conceptual dependency representation.

The Primitives

According to Schank, the primitive ACTs were decided upon by "noticing structural similarities that existed when sentences were put into an actor-action-object framework" (Schank 1975: 34). Schank points to six ACTs that are 'major' ACTs. They are:

1. ATRANS - give, take and buy
2. PTRANS - go and fly
3. PROPEL - push, pull and kick
4. MTRANS - remember, forget, tell and read
5. MBUILD - describe, decide, imagine, consider and answer
6. INGEST - eat, breathe, shoot and smoke

Schank states that all conceptualizations require conceptualizations that are instruments for them. The remaining primitive ACTs are primarily used for the instrumental conceptualizations (Schank 1975: 35).

I would consider these to be 'supporting' ACTs as they occur in a supporting role to the six aforementioned ACTs.

1. GRASP - clutch
2. ATTEND - listen
3. SPEAK - say
4. MOVE - kick
5. EXPEL

Verbs Spanning Multiple ACTs

Schank uses 'take' as an example of a verb that qualifies for two ACT categories:

He took an aspirin - INGEST.

John took the book from Mary - PTRANS

What I believe is that the two senses of 'take' will fall into different categories not only for the ACTs but for the verb categories in Levin's book. Levin provides some example sentences, unlike Schank, and hopefully the syntactic examples will point to potential case frames. Also, the example sentences may point to the constraints that exist on the slot fillers for the different ACTs.

Overview of Conceptual Dependency Section in Rich's "Artificial Intelligence"

The section in this book is a nice overview of conceptual dependency. The highpoint of the description is a listing of the fourteen rules of conceptual dependency including examples of English sentences for each of the rules. Other papers make mention of the rules or give examples but none are quite as tidy as Rich's representation. Unfortunately, the section provides no further detail that is not covered by other articles already reviewed.

The book makes reference to two books that may be of interest:

1. Schank, Roger 1975. *Conceptual Information Processing*. Amsterdam, North-Holland

2. Schank, Roger and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding*. Hillsdale, NJ:Earlbaum

Review of Levin's *English Verb Classes and Alterations*

I reviewed Levin's book *English Verb Classes and Alterations*. The first part of the book (it is broken into two parts) is devoted to 'diathesis alterations'. The alterations are subdivided into groups based upon syntactic case frames (Levin 1993: 22). There are three types of alterations:

1. Transitive alterations
2. Alternate expressions of arguments that do not affect transitivity, and
3. Alterations that arise when verbs permit "oblique" subjects (Levin 193: 22)

The alterations are important because they break verbs into groups based upon a sentence's syntax. The possible choices for the unknown verb in the following sentence:

I unknown-verbed at/on/against the table.

Is restricted to a single group of related verbs simply due to the fact that the prepositional phrase is headed by the preposition *at*. Grouping verbs based upon sentence syntax relies heavily upon the combination of prepositions used, the permissible alterations of the noun phrases (NPs) of the prepositions and the class membership or type of the NP. The alterations will not allow verbs to be placed directly into ACT categories but are very promising in that they allow for verb categorization based upon the syntax of the sentence alone.

This paper describes alterations using two example verbs, 'give' and 'take'. The analysis details how a sentence with an unknown verb could be validated against numerous, predefined rules to determine possible alteration group membership. From there, another set of rules could be applied to determine which of the ACT categories the unknown verb belonged to.

I choose the verbs 'give' and 'take' because they are the two traditional examples used by texts describing conceptual dependency. My original premise was that a verb like 'give' would map to a couple alterations. From there, each alteration would map to a few ACTs and the problem of verb categorization would be solved. This turned out roughly to be the case though the paper points out some difficult issues. The syntactic rules proposed by Levin's diathesis alterations, however, are very promising and provide a set of concrete, unchanging rules that can be used to categorize unknown verbs.

Overview of 'Give'

'Give' is the classic example of a PTRANS in conceptual dependency, though the verb spans several ACT categories. Levin labels 'give' as a *change of possession* verb. Other 'give' verbs are *feed, lease, lend, loan, pass, pay, peddle, refund, render, rent, repay, sell, serve* and *trade*. The two alteration types for 'give' verbs are causative/incohative and dative alteration. The dative alteration is of interest as it presents a syntactic case frame around which a rule can be founded.

Dative Alteration of 'Give'

The dative alteration is characterized by "an alternation between the prepositional frame 'NP1 V NP2 *to* NP3' and the double object frame 'NP1 V NP3 NP2'" (Levin 1993: 47). The NP found in the *to* prepositional frame can substitute as the first object in the double object construction. The following would be examples of sentences using the dative alteration for 'give':

John gave the car to Phil.

John gave Phil a car.

John gave a car.

John gave the car sitting in his front yard to Phil.

Additionally, the 'give' dative alteration has the constraint that there is an animacy restriction on the goal phrase (Levin 1993, 48). The target of the *to* prepositional phrase has the characteristic of being animate. In the previous examples this is clear because 'Phil' is a human. Levin argues that the animacy quality extends to organizations and companies. So, if 'Phil' were replaced with the word 'church', the target of the *to* prepositional phrase would still be viewed as animate.

Formulating a Rule Using 'Give', the Dative Alteration and Conceptual Dependency

A rule for using the dative alteration and 'give' might go something like this:

If unknown verb has the following alterations (here, V stands for the unknown verb):

1. **x V the y to z and/or**
2. **x V z the y**

and z has the characteristic of being animate,

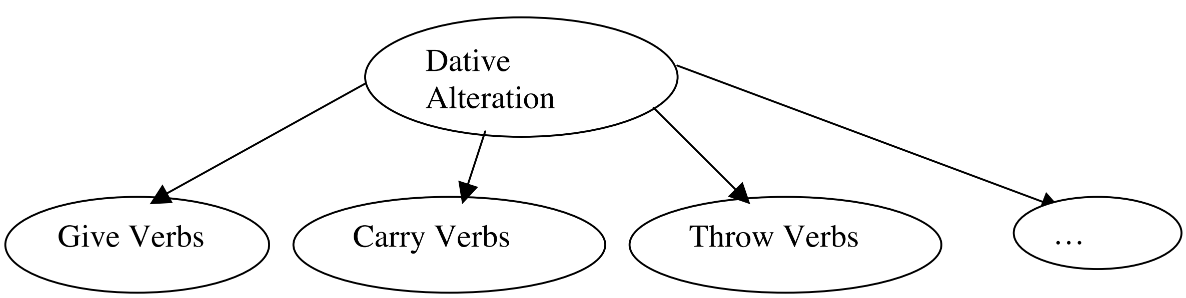
then V might be a 'give' verb in Levin's terms

As a further refinement a second clause can be added to the rule to determine which ACT case 'give' is being used with:

If the object being given is a physical object,
then 'give' is used in the PTRANS sense
//Example: *I gave the ball to Phil.*
Else, if the object being given is abstract,
then 'give' is used in the MTRANS *sense*
//Example: *I gave Phil an idea.*

Unresolved Issues Stemming From This Approach

The first problem deals with verbs from a different class than the 'give' class qualifying for the rule above. This is possible because 'give' class is not the only class of verbs to qualify for the dative alteration as the diagram below shows:



The hierarchy shows that multiple verb classes fall under the dative alteration, not just 'give'.

There may be ways to tweak the rule so that it places the unknown verb into the correct verb class. I have not fully explored this avenue as it would require a review of all the classes that fall under the dative case and determining what, if any, restrictions apply to the alteration of the NPs or if there are characteristics for the NPs inherent to certain verb classes. It is certainly the case that restrictions on the alterations exist for certain verb classes.

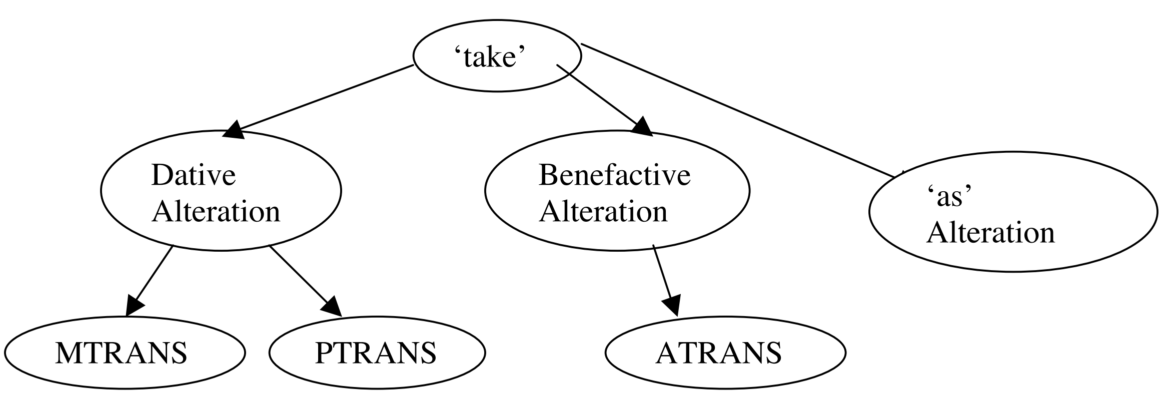
Another problem would be a sentence like the following:

I gave a kick to the ball.

It fails the animacy constraint that Levin applies to the goal of the *to* prepositional phrase of the dative alteration. I fear that no matter what the rule, exceptions will always exist.

Overview of 'Take'

'take' is a very complex verb with numerous alterations. I will not attempt to formulate a set of rules for the verb 'take' as I did for 'give'. Rules exist for 'take' that use the syntax of the target sentence, primarily focusing on the prepositions used and characteristics of the NPs. I worked out a rule for 'take' and it has basically the same form as the one for 'give'; first determine the alteration type, second, based on the characteristics of the NPs determine the ACT type. What is of interest for 'take' is the following hierarchy:



It would seem that the alterations could be another way to determine which class a verb is in. For instance, if the verb is in the dative, benefactive and 'as' alteration classes then it might be of the 'take' class. This would require numerous examples of the unknown verb and may not be realistic.

Conclusion

The alterations provide a comprehensive set of syntactic rules that will allow for categorization of unknown verbs into classes. The syntactic rules may not allow for direct inclusion in the ACT categories themselves but it is possible to assign ACT categories to unknown verbs by reviewing the class membership of the NPs.

Example Verbs for ACT Categories

Below is a table of the ACT categories and some example verbs from each. Ideally, I would like to create rules for each of the example verbs based upon Levin's alternations and my own observations of the verbs, especially observations from a conceptual dependency point of view.

ACT Category	Example Verbs
ATRANS – Transfer an abstract relation such as possession, ownership or control	Give, take, buy and pry
PTRANS - Transfer of physical location of an object	Go and fly
PROPEL – The application of force to an object	Push, pull and kick
MOVE – The movement of a body of an animal	Kick
GRASP – The grasping of an object by an	Clutch

actor	
EXPEL – The expulsion from body of an animal into the world	Spit
MTRANS – The transfer of mental information between or within an animal	Remember, forget, tell and read
MBUILD – The construction by an animal of new information from old information	Describe, decide, imagine, consider and answer
ATTEND – The action of directing a sense organ towards an object.	Listen
SPEAK - The action of producing sounds from the mouth	Say
INGEST – The action of an animal taking a substance into its body	Eat, breathe, shoot and smoke

Listing of Rules for Example Verbs

I created rules using the following process:

1. Go to the example verb and find all of Levin's alternations for the verb. Assume that each alternation is a rule unto itself.
2. Review a single alternation for the example verb fully. Create a Boolean type frame for the rule, essentially slots that will evaluate to either true of false. Try and keep the Boolean categories the same for the different alternations and verbs so it will be easier to compare them.
3. Look at some example sentences and get a feel for which ACT category the alternation most closely resembles. In some instances, alternations will span multiple ACTS but in others it will clearly only apply to a single ACT.

I created a generic frame for rules that roughly (there are some exceptions) each follows:

Verb Name – the verb's name
Rule Number – the rule number
Alternation Name – alternation name including Levin book section number
Description – brief overview of the alternation
Group Membership – group the verb belongs to including all related verbs
Other Groups – the listing of other groups that follow this rule. These groups sit conceptually at the same level as the 'Group Membership' group from above.
Transitivity – which transitive cases can exist for the alternation
Prepositional Frame – which prepositional frames exist
Alternation Restrictions – restrictions on alternation combinations
Semantic Constraints on Fillers – constraints on the slot fillers
Conceptual Dependency Categories – usually accompanied with example sentences

‘Give’ Rules

Verb Name: Give
Rule Number: 1
Alternation Name: Causative/Incohative (Levin 1.1.2.1)
Description: “Involve verbs with transitive and intransitive uses, where the transitive user of a verb V can be paraphrased as roughly “cause to V-intransitive”(Levin 1993: 27.)” This is a very broad alternation covering many verbs.
Group Membership: Give Verbs – *feed, give, lease, lend, load, pass, pay, peddle, refund, render, rent, repay, sell, serve* and *trade*
Other Groups: Roll, Break, Verbs of Change of State, Amuse-Type Psych-Verbs, Verbs of Change of Possession (Give, Contribute and Verbs of Future Having), Verbs of Cutting, Verbs of Contact by Impact, touch Verbs, Verbs of Killing, Destroy Verbs, Verbs of Appearance, Disappearance and Occurrence
Transitivity: Transitive and Bitransitive. For *give* verbs, the intransitive case does **not** apply
Prepositional Frame: Uses the *to* case frame with the following alternation:

I gave the book *to* Mary.
I gave Mary the book.

Alternation Restrictions: The following alternation would not be allowed:

The book gave to Mary.

Semantic Constraints on Slot Fillers:

Object – some type of physical object
Goal of preposition – has animate qualities

Conceptual Dependency Categories: PTRANS if viewed as exclusively as a change of possession

Verb Name: Give
Rule Number: 2
Alternation Name: Dative (Levin 2.1)
Description: Does not involve a change in transitivity of the verb. All the alternations are displayed by transitive verbs that take more than one internal argument and arise because these verbs allow more than one way of expressing these arguments (Levin 1993: 47).
Group Membership: Give Verbs – *feed, give, lease, lend, loan, load, pass, pay, peddle, refund, render, rent, repay, sell, serve* and *trade*
Other Groups: Alternating (includes Give, Verbs of Future Having, Bring and Take, Slide, Carry, Drive, Verbs of Throwing, Verbs of Transfer of a Message, Verbs of Instrument of Communication), Non-Alternating *to* Only (includes Primary Latinate, Say Verbs, Verbs of Manner of Speaking, Verbs of Putting with a Specified Direction, Verbs of Fulfilling) and Non-Alternating Double Object Only (includes Bill Verbs, Appoint Verbs, Dub Verbs, Declare Verbs)
Transitivity: Transitive and Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *to* case frame with the following alternation:

I gave the book *to* Mary.
I gave Mary the book.

Alternation Restrictions: The following alternation would not be allowed:

The book gave to Mary.

Semantic Constraints on Slot Fillers:

Object – some type of physical object
Goal of preposition – has animate qualities. For instance if the sentence were:

I gave the money to the United Way

then the *United Way* would be viewed as a living, animate organization.

Conceptual Dependency Categories: PTRANS if viewed exclusively as a change of possession. Not sure about the following two, though I think they ought to fit somewhere.

MTRANS if viewed as dealing with an abstract idea

I gave him the idea.

PROPEL if object is propel action

I gave the ball a kick.

I gave a push to the sled and down the hill they went.

'Take' Rules

Verb Name: Take
Rule Number: 1
Alternation Name: Dative (Levin 2.1)
Description: Same as Give Dative
Group Membership: Bring and take
Other Groups: Same as Give Dative
Transitivity: Transitive and Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *to* case frame with the following alternation:

I took the lunch *to* Mary.

I took Mary the lunch.

Alternation Restrictions: The following alternation would not be allowed:

The lunch took to Mary.

Semantic Constraints on Slot Fillers:

Object – some type of physical object

Goal of preposition – has animate qualities. For instance if the sentence were:

I took the money to the police department

then *police department* would be viewed as a living, animate organization.

Conceptual Dependency Categories: PTRANS if viewed as exclusively as a change of possession. Interestingly, could not think of examples of MTRANS or PROPEL which was the case for the *give* dative alternation.

Verb Name: Take
Rule Number: 2
Alternation Name: Benefactive (Levin 2.2)
Description: Closely resembles the dative alternation and is subsumed under it. It differs from the dative in involving the *for* preposition rather than the goal preposition *to*. Typically deals with verbs of creating or obtaining (Levin 1993: 48.)
Group Membership: Steal Verbs – *adduct, cadge, capture, confiscate, cop, emancipate, embezzle, exorcise, extort, extract, flich, flog, grab, impound, kidnap, liberate, lift, nab, pilfer, pinch, pirate, plagiarize, purloin, recover, redeem, reclaim, regain, repossess, rescue, retrieve, rustle, seize, smuggle, snatch, sneak, sponge, steal, swipe, take, thieve, wangle, weasel, winkle, withdraw* and *wrest*
Other Groups: Alternating (includes Build Verbs, Create Verbs, Verbs of Performance, Get Verbs), Non-Alternating *for* Only (includes Obtain Verbs, Verbs of Selection, Create Verbs and Steal Verbs)
Transitivity: Must be of Transitive or Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *for* case frame:

I took the book *for* Mary.

Alternation Restrictions: The following alternation would **not** be allowed:

I took Mary the book.

as it loses the benefactive concept of having taken a book solely for the purpose of giving it to Mary.
Semantic Constraints on Slot Fillers:

Object – some type of physical object

Goal of preposition – has animate qualities. For instance if the sentence were:

I took the money *for* the United Way

then the United Way would be viewed as a living, animate organization.

Conceptual Dependency Categories: ATRANS if viewed as exclusively as a change of ownership. For example:

I took the book from Greg for Mary.

Verb Name: Take
Rule Number: 3
Alternation Name: 'As' alternation (Levin 2.14)
Description: Found where there are transitive verbs that take complements predicated of their direct objects (Levin 1993: 78.)
Group Membership: Characterize Verbs – *accept, address, appreciate, bill cast, certify, characterize, choose, cite, class classify, confirm, count, define, describe, diagnose, disguise, employ, engage, enlist, enroll, enter, envisage, establish, esteem, hail, herald, hire, honor, identify, imagine, incorporate, induct, intend, lampoon, offer, oppose, paint, portray, praise, qualify, rank, recollect, recommend, regard, reinstate, reject, remember, repressed, repudiate, reveal, salute, see, select, stigmatize, take, train, treat, use, value, view* and *visualize*.
Other Groups: Appoint Verbs, Non-Alternating *as* Only (Characterize Verbs), Non-Alternating Double Object Only (Dub Verbs, Declare Verbs, Bill Verbs)
Transitivity: Must be of Transitive or Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *as* case frame:

I took the insult *as* a compliment.

Alternation Restrictions: The following alternation would **not** be allowed:

I took the insult a compliment.

Semantic Constraints on Slot Fillers:?
Conceptual Dependency Categories: MTRANS - This would seem to be the only ACT as the *as* preposition is very restrictive in this sense.

Verb Name: Take
Rule Number: 4
Alternation Name: Location Subject Alternation (Levin 3.6)
Description: Takes oblique subjects that can be characterized as locations. The location subjects are used to describe the capacity of the location with respect to the action named by the verb (Levin 1993: 82.)
Group Membership: Alternating Verbs – *Carry, contain, fit, feed, hold, house, seat, serve, sleep, store, take* and *use*.
Other Groups: None
Transitivity: Must be of Transitive or Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *in* case frame:

The hotel takes two people *in* each room.
Each room takes two people.

Alternation Restrictions: None
Semantic Constraints on Slot Fillers: The goal of the *in* preposition must be a location.
Conceptual Dependency Categories: Possibly ATRANS. For example:

Each room *takes (ownership of)* two people.

Verb Name: Take
Rule Number: 5
Alternation Name: There Insertion (Levin 6.1)
Description: Very restrictive case. Used for *took (place/shape)* type sentences. Not of much use to a verb algorithm unless the unknown verb is took. Would be trivial to create a rule for this.

Verb Name: Take
Rule Number: 6
Alternation Name: Bound Non-reflexive Anaphor as Prepositional Object (Levin 7.7)
Description: Very restrictive case. Deals with sentences where a pronoun in the prepositional phrase may or must be understood to be a coreferent to the subject (Levin 1993: 104.) Examples:

Jane brought the book *with her*.

Would be easy to code a rule for this but it specifically applies only to the verbs *give* and *take* so, how useful it would be is debatable.

'Pry' Rules

Verb Name: Pry
Rule Number: 1
Alternation Name: Clear Alternation of Locative Alternation (Levin 2.3.3)
Description: Found with certain verbs that relate to putting substances on surfaces or things in containers or to removing substances from surfaces or things from containers (Levin 1993: 52).
Group Membership: Remove Verbs – *abstract, cull, delete, discharge, disgorge, dislodge, dismiss, disengage, draw, eject, eliminate, eradicate, evict, excise, excommunicate, expel, extirpate, extract, extrude, lop, omit, ostracize, oust, partition, pry, reap, remove, separate, sever, shoo, subtract, uproot, winkle, withdraw* and *wrench*.
Other Groups: Clear Verbs, Non-Alternating *from* Only (includes Remove Verbs, Banish Verbs and Steal Verbs), Non-Alternating *of* Only (includes Cheat Verbs)
Transitivity: Transitive and Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *from* frame:

The thief pried the safe *from* the wall.

Alternation Restrictions: The following alternation would not be allowed as pry with the *from* preposition is specifically non-alternating:

The thief pried the safe of the wall.

Semantic Constraints on Slot Fillers: *from* target is a location
Conceptual Dependency Categories: ATRANS

Verb Name: Pry
Rule Number: 2
Alternation Name: *Apart* Reciprocal Alternation Transitive (Levin 2.5.3)
Description: Here, *apart* is viewed as indicating the resulting configuration. The prepositions involved are *from, out (of),* and *off (of)* (Levin 1993: 62.) The following example sentences make this clear.
Group Membership: Split Verbs – *blow, break, cut, draw, hack, hew, kick, knock, pry, pull, push, rip, roll, saw, shove, slip, split, tear, tug* and *yank*.
Other Groups: Alternating Verbs (Split Verbs), Non-Alternating Verbs (includes Separate Verbs and Disassemble Verbs)
Transitivity: Transitive and Bitransitive. The intransitive case does **not** apply
Prepositional Frame: Uses the *from, out (of),* and *off (of)* frames:

The thief pried the hood ornament *off of* the Jaguar.
The dentist pried the tooth *out of* the jaw.

Alternation Restrictions: Alternations are allowed as long as the *apart* reciprocal is introduced:

The boy pried the Legos *apart*.

Semantic Constraints on Slot Fillers: Possibly some constraints having to do with a part/whole relationship? Nothing jumps out either than the tow items being pried apart had the characteristic that they were conjoined prior to being pried apart.

Conceptual Dependency Categories: ATRANS - The person who does the act of *prying* gains ownership of the object that has been pried off.

Verb Name: Pry

Rule Number: 3

Alternation Name: Simple Reciprocal Intransitive (Levin 2.5.4)

Description: An extremely limited alternation as is the case with most intransitive alternations. Essentially states it is possible to have the sentence:

Bill and Mary *pried*

but not

Bill *pried* with Mary

Verb Name: Pry

Rule Number: 4

Alternation Name: *Apart* Reciprocal Alternation (Levin 2.5.5)

Not complete yet ☹

Overview of "Inferring the Meaning of Verbs from Context"

The paper reviews the following three methods for acquiring the meaning of an unknown verb from context:

1. A computational system that acquires the meaning of an unknown verb from context
2. A statistical analysis of the predictive features of a verb based on context.
3. Experiments run on adults to guess the meaning of missing verbs from context (Weimer-Hastings 1998: 1).

The first point made by the paper is that verbs are inherently harder to learn than are nouns. They make the point that for a verb such as *hijacked*, there exists semantic constraints on the slot-fillers that give limitations as to what the nouns might possibly mean (Weimer-Hastings 1998: 1). Because constraints are typically associated with the verbs and not the nouns, learning unknown verbs is much harder task. The paper goes into a description of Camille, a cognitive agent, and the functioning of its underlying verb algorithms. At the core of Camille's processing systems is a comparison function that seeks to compare each slot-filler with semantic constraints (Weimer-Hastings 1998: 2). Syntactic case frames are used but only indirectly; Camille is not equipped with a comprehensive set of rules that directly analyze syntactic structure of the entire sentence. This is something the research I am pursuing hopes to create.

Included is an example rule for the verb *deny*. Like the rules I hope to create based on Levin's alternations, the example rule is composed of a set of Boolean features that unknown verbs are run against.

The paper concludes with the following comment that sheds light on the importance of syntax to the entire process of giving meaning to an unknown verb: "One possibility (why humans are better at guessing the meaning of a verb than is Camille) is the syntactic context of the sentence, which Camille did not use, but a statistical corpus analysis showed could contribute significantly to verb inference." This single sentence, along with my continuing research, leads me to believe that syntactic structure is an essential clue in applying a meaning to an unknown verb.

References

1. Hardt, S.L. (1987), "Conceptual Dependency", in Stuart Shapiro and David Eckroth (eds.), *Encyclopedia of Artificial Intelligence* (New York: John Wiley and Sons 1987): 194-195
2. Levin, Beth (1993), *English Verb Classes and Alternations* (Chicago: The University of Chicago Press).
3. Lytinen, Steven L. (1992), "Conceptual Dependency and its Descendants", in Fritz Lehman (ed.), *Semantic Networks in Artificial Intelligence* (Oxford: Pergamon Press): 51-73.
4. Lytinen, Steven L (1986), "Dynamically Combining Syntax and Semantics in NLP", *Proceedings of the Fifth National Conference on Artificial Intelligence*, (Philadelphia): 574-578
5. Schank, Roger (1972), "Conceptual Dependency: A Theory of Natural Language Understanding", *Cognitive Psychology* (3)4: 552-631

6. Schank, Roger; Rieger, Charles J. (1974), “Inference and the Computer Understanding of Natural Language”, in Ronald J. Brachman and Hector J. Levesque (eds.), *Readings in Knowledge Representation* (Los Altos: Morgan Kaufman Publishers): 120-139
7. Schank, Roger (1975) “Theoretical Issues in Natural Language Processing: The Primitive ACTs of Conceptual Dependency”, *ACL Anthology* http://acl.ldc.upenn.edu/T/T75/T75.2008.pdf
8. Schank, Roger (1975), *Conceptual Information Processing* (North-Holland, Amsterdam: American Elsevier)
9. Rich, Elaine (1991), *Artificial Intelligence* (McGraw-Hill Inc.: New York), 277-284
10. Rieger, C. (1972), "Conceptual Memory and Inference", *Conceptual Information Processing* (North-Holland, Amsterdam: American Elsevier)
11. Riesbeck, C. (1972), "Conceptual Analysis", *Conceptual Information Processing* (North-Holland, Amsterdam: American Elsevier)
12. Weimer-Hastings, Peter; Graesser, Arthur C.; Weimer-Hastings, Katja (1998), "Inferring the Meaning of Verbs from Context" (The University of Memphis), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*

Proposed Enhancements to Ehrlich's Verb Algorithm

Justin M. Del Vecchio

Summer 2003

Independent Study

Abstract

This paper proposes enhancements to the current version of Ehrlich's verb algorithm based on an examination of 27 sample verbs and their accompanying verb alternations as determined by Beth Levin (1993). A verb alternation is syntactic behavior that is common to a group of verbs. For example, rummage verbs, which fall under the *Search* Alternation, commonly use the prepositions *through* and *for* in a particular order; *I rummaged through the desk for a pen*. The proposed enhancements take the verb alternations of the 27 sample verbs and restate each as a rule that an unknown verb can be verified against. Each rule examines the syntactic and semantic content of a sentence to determine if it can be assigned to a specific verb alternation. Most of the verb alternations detailed by Levin use both syntactic and semantic behavior for their definition with a clear emphasis on syntactic behavior. Included are discussions of the new and existing case frames that the enhancements would work with. The 27 sample verbs were chosen from eleven of Schank's conceptual dependency categories to allow for a wide distribution of verbs.

Theory Behind the Enhancements

Many verb alternations can be viewed as a syntactic/semantic fingerprint that identifies a group of verbs. Consider the made-up verb *furged* in the sentence below:

John furged the hood ornament off of the car.

A possible syntactic clue from the above sentence might be the prepositions *off of*. Levin repeatedly demonstrates that prepositions can serve as syntactic clues that help categorize verbs into groups. As it turns out, sentences that use the prepositions *off of* as a precursor to the indirect object typically use a verb from one of the following groups (Levin 1993: 62):

Apart Reciprocal Alternation

Alternating Verbs:

- Split Verbs: blow, break, cut, draw, hack, hew, kick, knock, pry, pull, push, rip, roll, saw, shove, slip, spit, tear, tug, yank

Non-Alternating Verbs:

- Separate Verbs: decouple, differentiate, disconnect, disentangle, dissociate, distinguish, divide, divorce, part, segregate, separate, sever
- Disassemble Verbs: detach, disassemble, disconnect, partition, sift, sunder, unbolt, unbuckle, unbutton, unchain, unclamp, unclasp, unclip, unfasten, unglue, unhinge, unhitch, unhook, unlace, unlatch, unlock, unleash, unpeg, unpin, unscrew, unshackle, unstaple, unstitch, unzip

By merely focusing on the prepositions *off of* and the fact that they immediately precede the indirect object, the unknown verb was narrowed down from being a member of the set of all verbs to being a member of a set consisting of 61 verbs.

Note that at the second highest level, *furged* can belong to two distinct groups of verbs: Alternating Verbs and Non-Alternating Verbs. Imagine there was an additional sentence using the unknown verb so the knowledge based consisted of:

John furged the twig and branch apart.

John furged the hood ornament off of the car.

These sentences would demonstrate the alternation with one using the preposition *off of* and the other not. The alternation characteristic of the verb *furged* would serve to further refine its group membership, implying it can only belong to Alternating Split Verbs. By examining the syntactic elements of two sentences, the made-up verb can be assigned to a group consisting of 20 verbs.

An open question is whether or not this group of 20 verbs can be cut down in size. For instance, would it be possible to take the set of verbs returned for the verb alternation and place subsets of the group into specific ACT categories? The benefit of being able to place the verbs into ACT categories and stating "The unknown verb is probably a PTRANS" is that it provides a much more precise definition than saying "The verb is either throw or toss or heave …". It would also bridge Levin's verb alternations with Schank's conceptual dependency categories. How to go about this is not clear at this time.

The 27 Sample Verbs

Table 1 contains the sample verbs whose verb alternations were examined.

ACT Category	**Example Verbs**
ATRANS – Transfer an abstract relation such as possession, ownership or control	Give, take and buy
PTRANS - Transfer of physical location of an object	Go and fly
PROPEL – The application of force to an object	Push, pull, pry and kick
MOVE – The movement of a body of an animal	Kick
GRASP – The grasping of an object by an actor	Grab
EXPEL – The expulsion from body of an animal into the world	Spit
MTRANS – The transfer of mental information between or within an animal	Remember, forget, tell and read
MBUILD – The construction by an animal of new information from old information	Describe, decide, imagine, consider and answer
ATTEND – The action of directing a sense organ towards an object.	Listen
SPEAK - The action of producing sounds from the mouth	Say
INGEST – The action of an animal taking a substance into its body	Eat, breathe, shoot and smoke

Table 1

The sample verbs were chosen so that representatives existed for each of Schank's eleven conceptual dependency (cd) categories although the algorithm enhancements do not specifically pertain to Schank's theory of cd. The pre-ordering based on cd categories was done to determine if a relationship exists between Levin's verb alternations and Schank's cd categories. The hope is that the two can be merged. Though some cd categories seem to correlate to specific verb alternations, there is not enough evidence to give a definitive answer at this time.

The Verb Alternations Covered by the 27 Sample Verbs

Levin defines 82 different verb alternations. After examining 27 sample verbs I was able to come up with rules that address 22 of these alternations. 17 of the alternations deal with examining specific prepositions and the semantic content of the sentence. Table 2 is a breakdown of these 17 prepositions and corresponding alternations. The reader will notice that not all of the 27 verbs are represented in this table. This is due to the fact that some verbs had no usable alternations (either the alternation was too broad or far too specific to be of any practical use) or the verb was not contained in Levin's book at all. This was the case for the following verbs: spit, forget, decide, answer, breathe

Preposition(s)	Initial Verb	Corresponding Alternation
To	Give, take, fly, push, pull, kick, tell, read, describe, imagine, consider, say, shoot	1.1.2.1, 2.1
For	Take, buy, grab, shoot	2.2, 3.9
As	Take, remember, describe, imagine, consider	2.14
In	Take	3.6

From	Pry, grab	2.3.2
off (of)	Pry, push, pull, kick	2.5.3, 2.5.6
Out (of)	Pry, push, pull, kick	2.5.3, 2.5.6
Directional prepositions through, around, onto, into, over	Fly, shoot	1.1.2.2 (minor), 1.4.1
Phrase 'their way' followed by a directional preposition: through, around, onto, into, over	Push	1.2.7
At	Push, pull, kick, eat, shoot	1.3
At/on/against	Push, pull, kick, eat, shoot	1.3
With	Kick, shoot	2.8
Against	Kick, shoot	2.8
For/in or for/through or in/for or in/through	Listen	2.11

Table 2

The remaining five alternations not represented used semantic information for verb alternation categorization and did not rely on the presence of a specific preposition or, in some cases, any preposition at all. Table 3 shows these alternations and the associated verbs.

Initial Verb	Corresponding Alternation
Push, pull, kick	7.7
Kick	8.2 and 1.2.2
Shoot	2.12
Eat	1.2.1

Table 3

It is interesting that Levin emphasizes the importance of syntax when outlining how alternations work, yet almost all of the verb alternations she defines utilize the semantic content of the sentence in their definition somehow. Table three contains links to alternations that are defined solely by a sentence's semantic content.

Representing Prepositions with Case Frames

Verb alternations rely heavily upon the examination of prepositions and the enhancements proposed in this paper are predicated on being able to determine which prepositions are used in a sentence with which noun participants (subject, object or indirect object). Therefore, it is necessary to determine how a sentence's prepositions will be represented using case frames.

There are three possible approaches to representing the following sentence (Rapaport 2003):

John gave a book to Mary

Approach One: Include no information about the preposition like Figure 1. This is how most representations are currently done in SNePS. It would be the job of the grammar to reconstruct the sentence and recognize that the preposition *to* needs to be inserted before Mary.

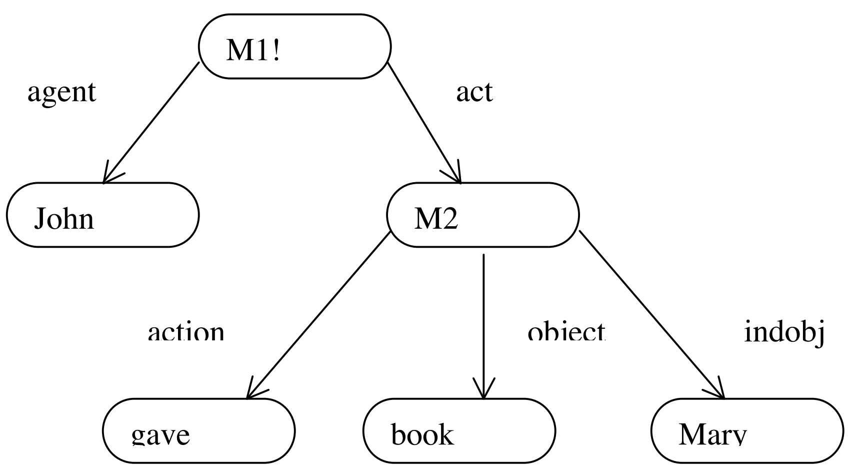


Figure One

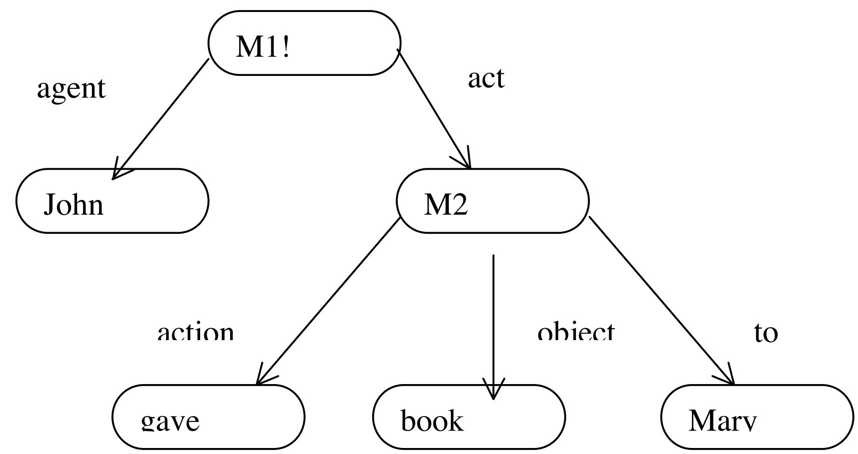


Figure Two

Approach Two – This approach would include information about the preposition directly in the arc label. In Figure 2, the *indobj* arc label is replaced by the *to* arc label.

Approach Three – This approach includes information about the preposition in the node itself. Figure 3 shows the preposition placed in a node and pointed to by a *prep* arc.

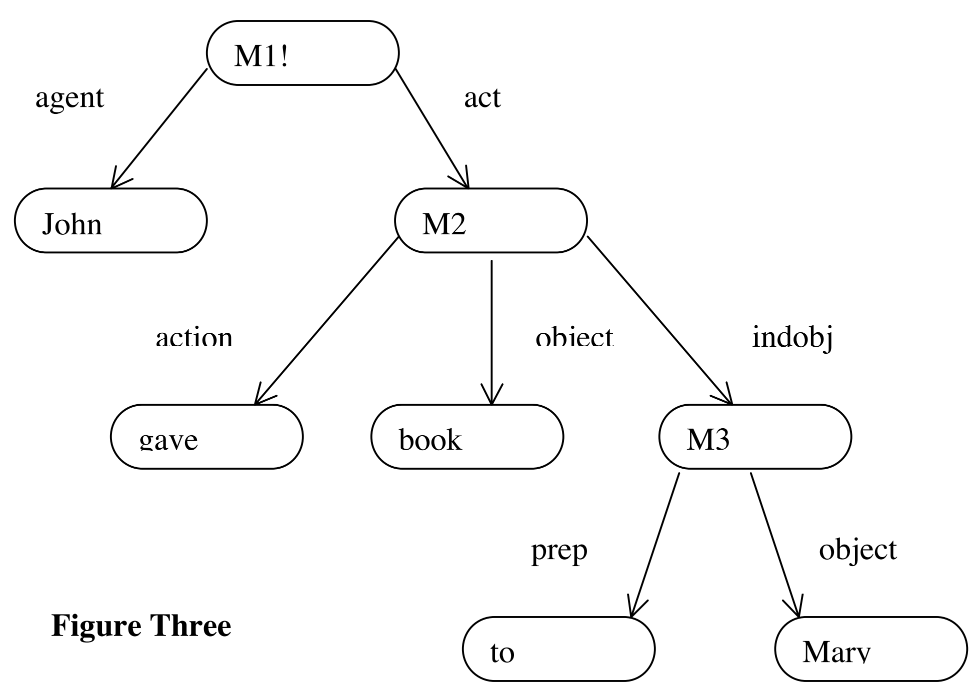


Figure Three

Below are some pros and cons for each approach:

- Con - Taking Approach One would require another programmer who is working on the grammar. This person does not exist, so Approach One is not viable at this time.
- Pro - Approach One would remove prepositions from the SNePS representation entirely. This might be a better approach to the problem as SNePS representations are not supposed to be language specific but instead universal in nature. Approach One would hold true to this tenet.
- Con - Approach Two is possible but would require numerous additional arc labels. For each preposition or preposition combination, an arc label would be necessary. This seems to be a high maintenance and does not hold in good form with SNePS representations.
- Pro - Approach Three places the information we are searching for inside a node. This will require less maintenance in the long run as there will be a single arc label, *prep*, to key on and any searching will be done on what that arc label points to. The algorithm will not have to be updated to handle new or changing arc labels.
- Con – Approach Three places prepositions squarely inside the SNePS representations. This makes them English specific instead of universal in nature.

Additional Representation Considerations

Levin does not use prepositions alone to define verb alternations. She also relies on the adverbs, adjectives and semantic content of a sentence when defining verb alternations. Below are all the non-preposition related representation issues I came across that need to be addressed. They follow no specific pattern. Their representation, like prepositions, needs to be agreed upon up-front so the verb algorithm knows what to search for.

1. If something has the characteristic of being a physical object. Numerous Levin alternations make use of this fact. A possible case frame could be Object/Property.

2. If something have animate qualities. Numerous Levin rules make use of this fact. A possible case frame could be Object/Property.
3. The adverb *apart* needs to be represented. This is necessary for a sentence like the following: *John pulled the brawling boys apart.* It applies to Levin rules 2.5.3 and 2.5.6. I do not know what a possible case frame might be.
4. If the subject is represented by the specific phrase *their way* where *their* is an adjective and *way* a noun. It applies to Levin rule 1.2.7. A possible case frame could be Mod/Head with Mod pointing to *their* and Head pointing to *way*.
5. If the subject is a sum of money. It applies to Levin rule 3.9. I do not know what a possible case frame might be.
6. If something has the characteristic of being is a physical location, say a room, bus, airplane, etc. It applies to Levin rule 3.6. A possible case frame could be Object/Property.
7. Sentences where there is a subject, an object and an indirect object with a prepositional phrase complement where a pronoun in the prepositional phrase is a coreferent to the subject. Basically sentences like: *I poured water on myself*. It applies to Levin alternation 7.7. One possible case frame could be Equiv/Equiv with one arc pointing to the subject and the other the indirect object.
8. If the object is a body part and is owned by the subject. Target sentences would be similar to the following: *Jennifer pursed her lips.* It applies to Levin alternation 8.3. A possible case frame could be Part/Whole.
9. If the indirect object is a body part and is owned by the object. A target sentence would be similar to: *John scratched the cat on the ear.* It applies to Levin alternation 2.12. A possible case frame could be Part/Whole.

Pseudocode Verb Algorithm Enhancements

Step One – Cycle through and find all entries for the unknown verb in the knowledge base. When an instance of the unknown verb is found, place the node that represents the sentence in an array. I believe that Ehrlich's current version of the verb algorithm does not store this information, instead creating the list each time it is needed.

Step Two

Cycle through each node in the list created in **Step One**. Do **Step Three** until the end of the list has been reached. Each time a result is generated for **Step Three**, cache the result.

Step Three

If the verb uses the preposition *to* for the indirect object:

> **If** the object is a physical object **and** the goal of the preposition has animate qualities
>
> **Then:**
>
>> **If** the following type of alternation exists where object and indirect object are swapped and the preposition *to* is not used:
>>
>> 1. I gave the ball *to* Mary.
>> 2. I gave John the ball.
>>
>> **Then:**
>>
>>> Assign the verb to the Dative Alternation – Alternating Verbs (Levin 1993: 47). This represents a very large number of verbs. A general observation would be that the verb represents some type of change of possession though this is not always the case. Return the following list of verb groups: Give Verbs, Verbs of Future Having, Bring and Take, Send Verbs,

Slide Verbs, Carry Verbs, Drive Verbs, Verbs of Throwing, Verbs of transfer of Message, Verbs of Instrument of Communication

Else:

Assign the verb to the Dative Alternation (Levin 1993: 47). This rule returns a very large number of verbs due to the fact the preposition *to* is so ubiquitous in its use. A general observation would be that the verb represents some type of change of possession though this is not always the case.

Return the following list of verb groups: Give Verbs, Verbs of Future Having, Bring and Take, Send Verbs, Slide Verbs, Carry Verbs, Drive Verbs, Verbs of Throwing, Verbs of Transfer of Message, Verbs of Instrument of Communication, Latinate Verbs, Say Verbs, Verbs of Manner of Speaking, Verbs of Putting With a Specified Direction, Verbs of Fulfilling

If the verb uses the preposition *for* for the indirect object:

If the subject is a sum of money

Then:

Assign verb to Sum of Money Subject Alternation (Levin 1993: 83). This will return a small number of verbs, 18. Return the verbs for the following groups of verbs: Verbs of Obtaining, Build Verbs

If the object of the sentence uses the preposition *in*

Then:

Assign the verb to the *Search* Alternation (Levin 1993: 70). This will return a small number of verbs, 31.

Return the verb groups: Hunt Verbs, Search Verbs

If the object of the sentence uses the preposition *through*

Then:

Assign verb to the *Search* Alternation (Levin 1993: 70).

Return the verbs: bore, burrow, delve, forage, fumble, grope, leaf, listen, look page, paw, poke, rifle, root, rummage, scrabble, scratch, snoop, thumb, tunnel

If the object of the sentence is a physical object **and** the goal of the preposition has animate qualities

Then:

If additional sentences exist that display the following alternation where object and indirect object are swapped and the preposition *for* is dropped:

1. Grandma knitted a sweater *for* Timmy.
2. Grandma knitted Jane a sweater.

Assign the verb to the Benefactive Alternation – Alternating Verbs (Levin 1993: 48). This is similar to the Dative Alternation and returns many verbs.

Return the following list of verb groups: Build Verbs, Create Verbs, Verbs of Performance, Get Verbs

Else assign the verb to the Benefactive Alternation (Levin 1993: 48). This is similar to the Dative Alternation and returns many verbs.

Return the following list of verb groups: Build Verbs, Create Verbs, Verbs of Performance, Get Verbs, Obtain Verbs, Verbs of Selection, Create Verbs and Steal Verbs

If the verb uses the preposition *as* for the indirect object:

If additional sentences exist where the indirect object does not use the preposition *as*:

1. The president appointed Talley *as* press secretary.
2. The president appointed Talley secretary of state.

Then:

Assign the verb as having the *As* Alternation - Alternating (Levin 1993: 78).

Return the following verbs: acknowledge, adopt, appoint, consider, crown, deem, designate, elect, esteem, imagine, mark, nominate, ordain, proclaim, rate, reckon, report, want

Else:

Assign the verb as having the *As* Alternation (Levin 1993: 78). It will return around 56 verbs.

Return the following list of verb groups: Appoint Verbs, Characterize Verbs

If the verb uses the preposition *in* for the indirect object:

If the object of the sentence uses the preposition *for*

Then:

Assign the verb to *Search* Alternation (Levin 1993: 70). It will return a group of around 30 verbs.

Return the verb groups: Hunt Verbs, Search Verbs

If the goal of the *in* preposition is a location **and** the object is modified by a quantity:

1. We seat twelve people in a bus.
2. The coach car houses two people in a compartment.

Then:

Assign the verb as having Location Subject Alternation (Levin 1993: 82).

Return the following verbs: carry, contain, fit, feed, hold, house, seat, serve, sleep, store, take, use

If the verb uses the preposition *from* for the indirect object:

If there exist sentences using the verb that use the preposition *of* for the indirect object so the following alternation exists:

1. Tammy cleared the dishes *from* the table.
2. Tammy cleared the table *of* dishes.

Then:

Assign the verb as having the *Clear* Alternation – Alternating (Levin 1993: 52).

Return the following verbs: clear, clean, drain, empty

Else:

Assign the verb as having the *Clear* Alternation (Levin 1993: 52). Return a rather large group of verbs, roughly 80 verbs.

Return the verb groups: Clear Verbs, Remove Verbs, Banish Verbs, Steal Verbs,

If the verb uses the preposition *of* for the indirect object:

If there exist no sentences that use the verb that use the preposition *from* for the indirect object like:

1. Tammy cleared the table *of* dishes.
2. Tammy cleared the dishes *from* the table.

Then:

Assign the verb as having the *Clear* Alternation – Non-Alternating *of* Only (Levin 1993: 52). This will return around forty verbs.

Return the verb group: Cheat Verbs

If the verb uses the preposition *off (of)* or *out (of)* for the indirect object

Then:

If an existing sentence using this verb ends with the adverb *apart* like:

1. Tom pulled the brawling boys *apart*.

Then:

Assign the verb as having the *Apart* Reciprocal Alternation (transitive) – Alternating Verbs (Levin 1993: 62).

Return the following verbs: blow, break, cut, draw, hack, hew, kick, knock, pry, pull, push, rip, roll, saw, shove, slip, split, tear, tug, yank

Else:

Assign the verb as having the *Apart* Reciprocal Alternation (transitive) (Levin 1993: 62).

Return the following groups of verbs: Split Verbs, Separate Verbs, Disassemble Verbs

If the verb uses the preposition *off (of)* or *out (of)* for the object

Then:

If an existing sentence using the verb ends with the adverb *apart*

Then:

Assign the verb as having the *Apart* Reciprocal Alternation (intransitive) – Alternating Verbs (Levin 1993: 65).

Return the following verbs: blow, break, draw, kick, knock, pry, pull, push, rip, roll, shove, slip, split, tear, tug, yank

Else:

Assign the verb as having the Apart Reciprocal Alternation (intransitive) (Levin 1993: 65). Return roughly 22 verbs.

Return the following groups of verbs: Split Verbs, Separate Verbs, Differ Verbs

If the indirect object or object uses a directional preposition like *over, around, onto, into, through, up, down, along,* etc. (Note, Levin does not give a comprehensive list of directional prepositions so this will need to be researched further.)

If the object is the specific phrase *their way*

Then:

Assign the verb to the *Way* Object Alternation (Levin 1993: 40).

Return the following verbs: press, push, shove

Else:

Assign the verb as having Locative Preposition Drop Alternation (Levin 1993: 43). There is a good possibility that further Boolean checks can be implemented so only Roll Verbs get returned when the preposition *around* is used, etc.

Return the following verbs for each verb group: Run Verbs, Verbs that are Vehicle Names, Roll Verbs, Verbs of Inherently Directed Motion

If the object uses the preposition *at*

If other sentences using the verb exist that use the prepositions *on* or *against* for the object

Then:

Assign the verb to the Conative Alternation – Push/Pull Verbs (Levin 1993: 41).

Return the verbs: draw, heave, jerk, press, pull, push, shove, thrust, tug, yank

Else:

Assign the verb to the Conative Alternation (Levin 1993: 41). This will return a large number of verbs, probably around 80.

Return the verbs for the following groups: Hit Verbs, Swat Verbs, Poke Verbs, Cut Verbs, Spray/Load Verbs, Push/Pull Verbs, Eat Verbs, Chew Verbs

If there is a subject, an object and a prepositional phrase indirect object where the goal of the prepositional phrase is a coreferent to the subject

Then:

Assign the verb to the Bound Nonreflexive Anaphor as Prepositional Object alternation (Levin 1993: 104). Return roughly 58 verbs.

Return the verbs for the following groups: Contain Verbs, Brain and Take, Carry Verbs, Push/Pull Verbs, Pour Verbs, Coil Verbs, Spray/Load Verbs

If the object is a body part and is owned by the subject

Then:

Assign the verb to the Inalienably Possessed Body-Part Object (Levin 1993: 107) or Understood Body-Part Object Alternation (Levin 1993: 34). Return a large number of verbs, probably around 80.

Return the verbs for the following groups: Wink Verbs, Crane Verbs, Floss Verbs, Braid Verbs, Hurt Verbs

If the indirect object uses the preposition *against*

If other sentences using the verb exist and have an indirect object that uses the preposition *with*

Then:

Assign the verb to the With/Against Alternation – Alternating Verbs (Levin 1993: 67).

Return the verbs: bang, bash, batter, beat, bump, butt, dash, drum, hammer, hit, kick, knock, lash, pound, rap, slap, smack, smash, strike, tamp, tap, thump, thwack, whack

Else

Assign the verb to the With/Against Alternation (Levin 1993: 67).

Return the verbs for the verb groups: Hit Verbs, Throw Verbs, Break Verbs.

If the indirect object uses the preposition *with*

If no other sentences using the verb exist where the indirect object uses the preposition *against*

Then:

Assign the verb to the With/Against Alternation (Levin 1993: 67).

Return the verbs for the verb groups: Hit Verbs, Swat Verbs, Spank Verbs, Poke Verbs

If the indirect object is a body part and the object is the owner of that body part

If instances of sentences using the verb exist that use a preposition in conjunction with the indirect object

Then:

Assign the verb to the Body-Part Possessor Ascension Alternation (Levin 1993: 71).

Return the verb groups: Touch Verbs, Verbs of Contact by Impact, Poke Verbs

Else

Assign the verb to the Body-Part Possessor Ascension Alternation (Levin 1993: 71).

Return the verb groups: Touch Verbs, Verbs of Contact by Impact, Poke Verbs, Break Verbs, Verbs of Cutting

If the object is unspecified as in the second of the following two sentences:

1. Jill ate the ice cream.
2. Jill ate.

Then:

```
Assign the verb to the Unspecified Object Alternation (Levin
1993: 33).  The verb is assumed to have an object that is typical
of the verb.
Return the following list of verbs: bake, carve, chop, clean,
cook, crochet, draw, drink, dust, eat, embroider, hum, hunt,
fish, iron, knead, knit, mend, milk, mow, nurse, pack, paint,
play, plow, polish, recite, sew, sculpt, sing, sketched, sow,
study, sweep, teach, type, sketch, vacuum, wash, weave, whittle,
write.
```

References

1. Levin, Beth (1993), *English Verb Classes and Alternations* (Chicago: The University of Chicago Press).
2. Rapaport, William J. (2003), Meeting on July 1st