# LEARNING WORD MEANINGS FROM EXAMPLES

Robert C. Berwick

MIT Artificial Intelligence Laboratory, Room 820
545 Technology Sq., Cambridge, MA 02139

## ABSTRACT

This paper describes work in progress on a computer program that uses syntactic constraints to derive the meanings of verbs from an analysis of simple English example stories. The central idea is an extension of Winston's (Winston 1975) program that learned the structural descriptions of blocks world scenes. In the new research, English verbs take the place of blocks world objects like ARCH and TOWKR, with frame-based descriptions of causal relationships serving as the structural descriptions. Syntactic constraints derived from the parsing of story plots arc used to drive an analogical matching procedure. Analogical matching gives a way to compare descriptions of known words to unknown words. The "meaning" of a new verb is learned by matching pan of the causal network description of a story precis containing the unknown word to a set of such descriptions derived from similar stories that contain only known words. The best match forges an assignment between objects and relations such that the unknown veib is matched to a known verb, with the assignment being guided by syntactic constraints. The causal network surrounding the unknown item is then used as a scaffolding to construct a network representing the use of the novel word in a particular context. Words (and their associated stories) that are "best matches" are grouped together into a similarity network, according to the match score.

## I WORD ACQUISITION AND DEFINITIONS

This paper describes an analogical matching system that attempts to learn the causal descriptions of new words. The end result is that there arc no "definitions" in the sense of necessary and sufficient conditions determining word meanings; rather, what is output is an interconnected set of descriptions of the actual use of words in context (under a particular theory of "context", namely, causal network structure). The use of analogical matching here should not be viewed as a necessary ingredient of the learning system, but rather one way to represent a program's knowledge about the world. In other words, story plots serve as a proxy for systematic understanding of how the world works, and by matching stories the program can determine that a novel situation will work like an old one.

The word learning program is also embedded into a larger system that can acquire new syntactic rules for Knglish, as described in (Berwick 1979 1980 1982). The word learning component uses the larger system's determination of the syntactic category of a new word and its predicate-argument structure (if a verb). These last two abilities arc based on the X-bar theory of (Jackcndon'1977) and a theory of syntax that assumes a strong principle of lexical transparency (roughly, that the semantic argument structure of a verb appears at all levels of representation). The overall system thus provides a working example of how syntactic and semantic constraints can interact to aid in the learning of language.

Conceptually, the underlying assumption is that the meaning of a word is determined by the role it plays in a causal network description of an event, and that similar words are those that play similar roles in the description of similar events. To take an example that will be discussed in detail in Section 2, consider the following scenario:

> Suppose we arc given two versions of the story of Macbeth, one reporting that "Macbeth murders Duncan" and the other that "Macbeth assassinates Duncan". Further suppose that "murder" but not "assassinate" is a known word. We should conclude that "assassinate" is most like "murder", since, comparing stories, it seems to us that "assassinate" plays the same role that "murder" docs in the Macbeth plot. We should also conclude that "assassinate" has political overtones, since we note that the Macbeth story includes such relations as "Macbeth wants to be king" and "Macbeth becomes king". Probing further, later stories should inform us that the uses of "assassinate" and "murder" are slightly different, since "murder" needn't carry that political connotations that "assassinate" docs.
> We should also be able to use the story of Hamlet to deduce the same kind of relationship between "murder" and "assassinate".

litis scenario illustrates the kind of learning to be modeled by the program. The actual retrieval and matching of candidate stories is considerably more complex than that described in (Winston 1980). Two separate levels of filtering, geared to linguistic and "script-like" constraints, are invoked before causal network matching is even attempted (largely on account of the computational cost of the matching-_proccdure).

## II A SCENARIO SHOWS THE PROGRAM IN ACTION

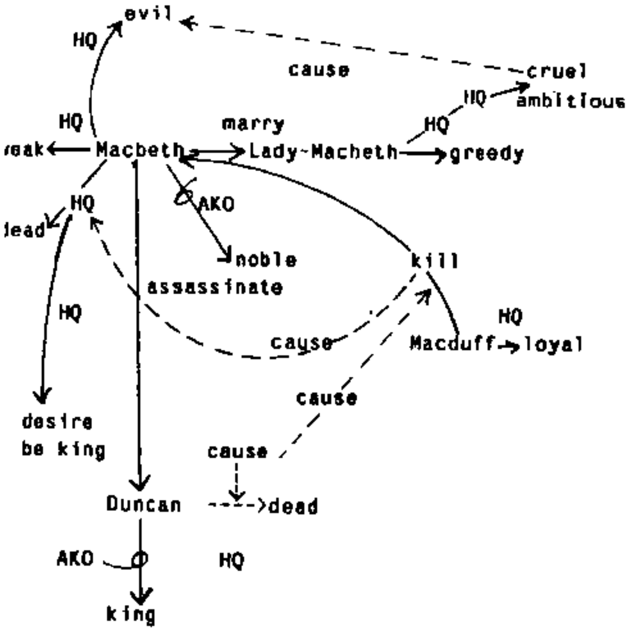
Perhaps the best way to gain some feel for the details of the acquisition procedure is to run through an example. Here is the actual input for the Macbeth story:

> MA is a story about Macbeth Lady-macbcth Duncan and Macduff. Macbeth is an evil noble. lady-macbeth is a greedy ambitious woman. Duncan is a king. Macduff is a noble, lady-macbcth persuades Macbeth to want to be king because she is greedy. Macbeth is

•This paper reports work done at the Artificial Intelligence laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is supported in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Reseaarch contract N0014-80-C-O505.

> evil because I.ad>-Macbeth is able to influence him and because Lady-Macbeth is greedy. She is able to influence him because he is married to her and because he is weak. Macbeth assassinates Duncan with a knife. Macbeth assassinates Duncan because Macbeth wants to be king and because Macbeth is evil. I ady-macbeth kills herself. Macduff is angry. Macduff kills Macbeth because Macbeth murdered Duncan and because Macduff is loyal to Duncan.

Using the techniques described in (Winston 1980) and (Katz and Winston 1982), the system builds a "causal network description" of the story, as shown below. This network is basically an object-oriented semantic network, with objects, agents, and qualities (and sometimes prepositional attitudes like beliefs or desires) serving as the nodes in the network and verbs serving as the links between nodes. The network is directed. It is dubbed "causal" because a major component is the addition of directed CAUSE links indicating that an object or relation at the link tail causally contributes to the relation or action at its head. (There arc some other special link labels. The *"\HQ"* link label denotes a two-place relation between nodes that is to be read as "X has quality Y"; AKO X = a-kind-of X. Some relations are added by one-step inference demons, described below; for example, "murder"--> adds "HQ dead". "I IQ evil" as appropriate. For reasons of clarity, some links in the description are omitted in the diagram below. There are also several evident weaknesses of the representation that will not be dealt with here. Most notably, aside from the implicit ordering imposed by causal relationships, there is no notion of temporality.)



Now suppose that a number of other stories have been previously analyzed and stored in causal network form, e.g., Macbeth. Hamlet, Julius Caesar, and the Taming of the Shrew, as discussed in (Winston 1980), and that none of these previous plot summaries used the word "assassinate."

The first job of the program is to successfully parse the input story containing the unknown word. Here, the new story differs in just two ways from the old story since the new word is used twice.

The techniques described in (Berwick 1982) (based on the Marcus parser (Marcus 1980) arc used to syntactically analyze the input:

(1) Consider the sentence, "Macbeth assassinates Duncan". Since all S's arc assumed to be of the form NP-VP, the parser predicts the existence of an NP Subject.
(2) With the NP disposed of, the parser now predicts that an Inflectional element and a Verb Phrase (VP) will be found. Now the constraint (sec (Jackendoff 1977)) that English Verb Phrases are headed by Verbs forces "assassinate" to be a Verb.
(3) Finally, "Duncan" is parsed as an NP and noted as an Object of the unknown verb. Since arguments arc assumed obligatory until proven otherwise by positive examples (see (Berwick 1982) for a discussion of this learnability constraint), this NP is presumed to be a required argument of "assassinate".

With the analysis of the novel sentences in hand, the remainder of the story analysis proceeds as before. A causal network representation can be built that is nearly identical to the network built previously, but with some key differences that flow from me lack of understanding of "assassinate." Most importantly, since there arc no one-step inference demons attached to "assassinate," the UQ relations of Duncan being dead and Macbeth being evil cannot be deduced. Still, the proper syntactic analysis permits the entire network to be built save for these two relations, since the connections between objects and actions are actually syntactic.

When the plot-unit portion of the system is complete, then the next step will be to output a a high-level description of the story in the style suggested by the work of (Lchncrt 1981). Note that the plot summary is intended to be a computational aid to story retrieval, and is not the central thrust of this work. Roughly, plot unit theory is grounded on a "chemical" model of representation. It assumes that the causal relationships in a story can be summarized by extracting out the "molecular structure" of the causal network. This is done by imposing a theory of plot molecular structure onto the more basic causal network. While there is no space to give the details of this procedure here, we simply note that given a network description, we can try to form a corresponding description in terms of "atomic" plot abstraction units like "loss" or "gain" (an agent experiences something negative or positive), this exercise is far from easy, however, and the rules for carrying out this translation are still under development.

Since the plot-unit step is still under development, it will not be further described here. Instead, what is currently done is to simply retrieve descriptions of a]] stories in the Shakespeare precis database. In our running example, these are the stories of Macbeth, Hamlet, Julius Caesar, and the Taming of the Shrew.

What next? The learning procedure takes the the causal network descriptions of each of the candidate stories and run the analogy matching program, pairing objects and relations in the new Macbeth story against objects and relations in each of the candidates until the best match is obtained. As mentioned, we restrict our attention to just that portion of the network that is immediately connected to the causal links emanating from "assassinate". This means that distantly related events and relations are not brought in for matching at all. (This metric of locality could of course be modified, but seems a reasonable first stab at a way to fix local context.) The matching program as designed can be primed to consider only matches that keep intact the objects on either end of the unknown "assassinate" link (this means that the match of "Macbeth" in Macbeth story #2 against "Duncan" in Macbeth story #1 is not even attempted, since this would be incompatible with the direction of the "murder" and "assassinate" links).

The match scores obtained are as follows. MA-2 is the new Macbeth story.

| | MA | HAM | JUL | SHR |
|---|---|---|---|---|
| MA-2 | 22 | 15 | 12 | 2 |

Not surprisingly, MA gets the highest match score, with the algorithm mating "assassinate" in MA-2 to "murder" in MA But the algorithm also pairs "assassinate" in MA-2 with "kill" in Hamlet and Julius Caesar, demonstrating that it would not be necessary to have an absolutely identical story in order to extract a useful word meaning. The algorithm weds "assassinate" to "love" in the laming of the Shrew, but this receives a low match score.

Next we modify the causal network description associated with "assassinate" according to the best match story. Any relations that locally associated with "murder" in the MA-1 network are added to the MA-2 description. In our example, this means that the "Duncan HQ dead" and "Macbeth HQ evil" links will be added to the MA-2 network. This modification corresponds to the assumption that since "assassinate" has been deduced as most similar to "murder", the inferences following from "murder" will also be true of "assassinate." (This of course is only approximately so, and later story examples where the target of assassination survives -- if explicitly indicated in the story -- would modify this link.)

Finally, the "meaning" of the novel verb is stored by associating with the word entry (i) a pointer to its causal network description; and (ii) a list of "nearest neighbor" words, according to the results of the matching algorithm. The similarity metric is normalized by the value obtained from matching an identical story against itself. In the case of multiple word matches (as with "kill", the best match word in both Hamlet and Julius Caesar), the highest scoring value is used. In the example above, the new entry for "assassinate" includes the related words of "murder" (score .88); "kill" (.60), and "love" (.08). (The score for MA is not 1 because the "HQ dead" and "IIQ evil" relations associated with "murder" arc not present in the initial MA-2 description.) The program also adds difference pointers indicating why the stored causal network description is different from each of the network descriptions associated with these three words. In the case at hand, there are no such differences, but if the best match story was, say, about a general homicide, then the "AKO" links would indicate that the characters in the "assassinate" story were all a-kind-of political figures, whereas those in the general homicide story need not be. Difference pointers could then be used to encode some of the distinctions between "murder" and "assassinate."

## III OPEN PROBLEMS

In this short paper, many important topics currently under investigation could not be covered. These include:

•- Better generalization schemes arc required. Currently, all details about a related story are kept. The only possible generalization is via the "AKO" hierarchy that drives the formation of difference pointers between word networks. This means that the choice of AKO vocabulary is crucial, since if the category' "noble" is not known then no generalizations about the category "noble" is possible. A better way of creating generalized categories out of old ones is needed, perhaps based on the work of (Keil 1979) on the development of semantic categories.

-- Plot unit summarization must be added to reduce the computational burden of matching against all known stories.

-- The use of causal network descriptions of stories to summarize world knowledge must be carefully examined.

## REFERENCES

[1] Berwick, R. "Learning Structural Descriptions of Grammar Rules From Examples," Proc. 5th IJCAI, Tokyo, 1979.

[2] Berwick, R. "Grammatical Analogues of Constraints On Grammars," Proc;. 18th Annual Meeting of the Association for Computational Linguistics, 1980.

[3] Berwick, R. Locality Principles and the Acquisition of Syntactic Knowledge. PhD Thesis, MIT Department of Electrical Engineering and Computer Science, 1982.

[4] Jackcndoff, R. X-har Syntax ; a Study of Phrase Structure. Cambridge, MA: MIT Press, 1977.

[5] Katz, B. and Winston, P. "Parsing and Generating English Using Commutative Transformations," MIT AI lab Memo, AIM 677, May 1982.

[6] Kcil, F. Semantic and Conceptual Development. Cambridge, MA: Harvard University Press, 1979.

[7] Lehncrt, W. "Plot Units and Narrative Summarization," Cognitive Science. 4 (1981).

[8] Marcus, M. A theory of Syntactic Recognition for Natural Language. Cambridge, MA: MIT Press, 1980.

[9J Winston, P. "Learning Structural Descriptions from Examples," in The Psychology of Computer Vision. McGraw-Hill, 1975.

[10] Winston, P."Lcarning and Reasoning by Analogy," CACM 23:12(1980).