Provost's Productivity Grant

The Intelligent Test Toolkit:
Story/Concept Generator

Final Report

Michael Yacci, Ph.D.
Associate Professor
Information Technology
RIT
Rationale

The history of technology in education has been littered with successes and failure. The successes are often the most conservative efforts. Often a new medium is simply substituted for another, such as computer-based training or instructional videos. However, to truly harness the power of the computer, we need to think about using the computer's capacity to represent knowledge and to manipulate symbols in a generative manner. This project is an early step in attempting to harness computer intelligence in an applied situation.

Concepts

A concept, as commonly defined in psychology, cognitive science, and instructional design, is a category used by humans to classify their world. A concept is a category name that groups unique items (or instances) and classifies them in the same category according to similarity in critical attributes or features.

To learn concepts, a student must see a variety of examples of the concept, so that the student can learn (a) which attributes are critical, (b) which attributes are irrelevant, and (c) to what degree does the concept generalize to "similar" cases. In instructional design terms, students are presented with examples and "non-examples," are asked to classify each instance, and are given feedback as to the correctness of their classification. (Merrill, 1982). While the exact cognitive mechanisms are not central to this paper, it is important to note that presenting a variety of concept instances is deemed crucial for learning to generalize a concept.

There are two types of concepts: concrete and defined (Gagne, 1978). Concrete concepts deal with real-life phenomena and are often learned without formal instruction (such as "dogs" and "cats"). Defined concepts are more typical of the classroom, and (as the name implies) usually have a formal definition that helps to define the category boundaries. One type of defined concept is referred to in this paper as a story/concept.

Story/Concepts

A story/concept is a concept that involves situation rather than an object. Many concepts in the social sciences involve more than the examination of a particular object--rather, they involve complex action events often with types of actors. These concepts are the basis of many social science principles. For example, the field of group dynamics uses the related concepts of constructive conflict and destructive conflict in the analysis of group interactions. However, these concepts can only be learned by pointing to many different examples of them--so that a student can learn to see the difference between them. While it may be possible to act through a variety of scenarios, many teachers will use "word problems" that explains a situation to a student and asks the student to classify the word problem.
Such a problem might look like following:

Bill and Jim are working on a project for the Chain Division of General Bicycle, Inc. Bill suggested that they use a titanium chain for the project, but Jim was against the idea. The dialogue went like this:

Bill: "I think the titanium will be more powerful--provide added strength."
Jim: "No--a thousand times no! Titanium will cost way too much."

Is this an example of constructive or destructive conflict? Why?

Similarly, by changing Jim's response, we have an example of destructive conflict.

Bill and Jim are working on a project for the Chain Division of General Bicycle, Inc. Bill suggested that they use a titanium chain for the project, but Jim was against the idea. The dialogue went like this:

Bill: "I think the titanium will be more powerful--provide added strength."
Jim: "No--a thousand times no! You have the worst ideas--always. Can't you think?"

Is this an example of constructive or destructive conflict? Why?

These story/concept examples are frequently descriptions of specific human events, used as practice or classroom examples to teach the concept.

Teachers and instructional designers who use story/concept examples in class, however, find that they are time consuming to create, and consequently may only provide a few examples for students to work with. For some students, a few examples may be sufficient, but others may need additional practice in classifying the concept. Therefore, a tool that could help teachers create a large number of story/concepts, quickly, would be useful.

The Intelligent Test Toolkit Story/Concept Generator

The Intelligent Test Toolkit Story/Concept Generator (SCG) is a natural language text generation tool, designed to take a single story/concept example, and to generate many parallel story/concept examples. Specifically, it (a) accepts user input, in the form of a paragraph, (b) prompts the user for the critical attributes of the story/concept and the degree of generalization and (c) creates as many parallel story/concept examples as possible, given its current knowledge base. These activities coordinate with three general components in the SCG. See Figure 1.
Figure 1. Components of the Story/Concept Generator

There are five screens within the SCG that represent the functions described above. Each screen of the SCG will be explained further. Note: the computational functions are described, not the exact programming structures.

1. **Parser Screen for User Input**
The user enters information into a Parser Screen that allows for free-text input. When the user has finished entering the story/concept example, the story is parsed in several different ways. First, each sentence from the story is parsed grammatically. In particular, the parser is seeking the verb in the sentence and also all *content words*. Harris (1985, pg 322) describes content words as nouns, verbs, adjectives, and adverbs (as distinct from *functional* words, such as articles and prepositions). Verbs are used to recognize the *conceptual primitive* (see Schank and Abelson, 1977) involved in the sentence and were not treated as content words in this system. Each primitive in the story is added to a *discourse structure* list, that keeps track of the order of the sentences. (A discourse structure is the basic abstracted template for the story--its skeleton.)

2. **Generalizability Parameter Screen**
The Generalizability Parameter Screen (GPS) is the first of two areas that are used to limit and understand the degree of generalizability of the story/concept. The GPS lists each content term in the story/concept. It then prompts the user to answer questions about the desired degree of generalization of each term.

All knowledge for the SCG is contained in a knowledge base. The knowledge base is a virtual semantic network representation of knowledge, using nodes and links (or relations) to connect them. Each knowledge proposition consists of two nodes and a link. Each is labeled. Nodes may each have a number of links to other nodes, but each link connects only two nodes.

The knowledge base is searched for each term listed on the GPS. This search looks only for hierarchical data (links labeled "a-kind-of"). The system assumes that each term is a
member of at least one broader category. For example, a doctor is also a kind of (AKO) worker. In some cases, terms may be members of several categories (a car is a kind of vehicle, is also a kind of gift, and is also a kind of status symbol). This form of "tangled" induction (Holland, Holyoak, Nisbett, & Thagard, 1986) could be modified with an inferencing system that makes a best guess as to context. However, such a system can be wrong; because this is a tool, it seemed most direct to ask the user for input to "untangle" then context. Therefore, in the case of "tangled" induction, the user is prompted to select the hierarchical category that suits the term in this story.

When the system encounters a proper name, it assumes that the proper name is irrelevant to the story. (This assumption is true in many cases. However, in some stories a particular proper name may be critical, such as "Einstein" or "Bill Clinton." Therefore it checks with the user.)

Any term that is not included in the knowledge base is stored and used exactly. In these situations, the user is simply informed so that he or she may add information to the knowledge base at a future date. The knowledge base contains an unlimited number of link relations. The determination concerning the best possible forms for knowledge base relations has yet to be worked on.

### 3. Semantic Network Screen

By labeling the categorical inheritance of terms on the GPS, the system can go directly to creating sentences, or can first further refine the generalizability parameters. Consider this sentence, with content terms underlined:

> The janitor used the broom to sweep the floor.

If we simply make categorical substitutions for content terms we could use any "worker" in place of janitor, any "tool" in place of broom, any "task" in place of sweep and any "object" in place of floor. However, these unconstrained substitutions can create the following, somewhat foolish sentences.

> The doctor used the paintbrush to wash the garbage
> The lawyer used the red pencil to take the windows
> The nurse used the seatbelt to water the patient

Obviously, the weakness is that certain key items have relationships that must be preserved in the new sentences for them to make sense. Because the discourse structure of a story/concept is made up of multiple related sentences, these relationships must be preserved and used throughout the new story/concept example.

However, in some cases these details may be irrelevant. Again, the control must be returned to the user to determine the criticality of preserving the relationships between terms.

A semantic network tool was created that allows users to create an ad hoc semantic net of the relationships that must be preserved as new story/concept examples are created. The
The semantic net provides a list of category names will be instantiated for the story and also a list of all links that are currently in the database. The user then creates the semantic net combining terms whose relationship must be preserved.

4. Semantic Network Two

One of the programming limitations to the semantic network tool was that all nodes needed to be connected to form a single network. However, there are cases in which there are two (or more) sets of relationships that are independent of each other, yet both sets must be preserved in the story/concept example. Note the following two-sentence story:

The janitor used the broom to sweep the floor. Bill moved the car to the garage.

In this story, the relationship between janitor, broom, and sweep must be preserved in the new stories. Additionally, the relationship between car and garage must also be preserved. But these two clusters of terms are basically independent of each other.

The doctor used a stethoscope to take blood pressure. Dave moved the boat to the dock.

Therefore, the SCG needed to provide a second semantic network to handle these additional relationships that are unrelated to the first semantic network.

5. Sentence Generator Output Screen

The Sentence Generator screen uses the decisions that were made on previous screens to re-construct a series of parallel story/concept examples.

Earlier, we stated that the verb was treated as a conceptual primitive rather than a content word. This means that each verb has information related to it (in a separate verb database) that explains the way the verb is conjugated, whether it is transitive or intransitive, how it is negated, and acceptable substitutions for the verb (see limitations, below).

The story/concept is reconstructed, sentence by sentence, using substitutions for its verb, and its content terms, constrained by the two semantic networks' preserved relationships. The sentence Generator will create a passive voice sentence in place of any sentence with a transitive verb, 50% of the time. (This number is selectable, is currently set to 50%.) Some substitutions produce an article mismatch ("a ice cube," "an book") and are corrected. Functional terms (prepositions, articles, conjunctive adverbs) are preserved.

If the user chooses to use the semantic network, the system attempts to create as many story/concepts as possible. This is often a fairly small number, given the constraints imposed upon the matching. However, when using totally unconstrained substitution, the system defaults to creating ten story/concept examples.
Assumptions of the Knowledge Base

The knowledge base is a critical component of the system, and was designed with specific assumptions in mind.

?? The knowledge base stores all knowledge as propositions: node-link-node relationships

?? The knowledge base does not store data about specific people. For example, the knowledge base stores the proposition that a janitor does the task of sweeping floors. It does not store the proposition that Dave does the task of sweeping floors.

?? The knowledge base stores knowledge of "real world" objects, job categories, and tasks. It does not store linguistic data.

?? Linguistic data is stored in two separate databases. A verb database contains information about verbs, such as whether they are conjugated in a regular manner, and how they are negated. A proper name database allows the system to recognize male or female proper names. A location name database should also be added in the future to recognize proper names of locations.

?? The knowledge base represents a large semantic network of knowledge that should be previously known to the SCG system. It is assumed that such a knowledge base would grow over time, as uses of the system add knowledge to it. (Note: There is no current provision for adding to the database, although such a feature is relatively easy to add.)

?? It is assumed that users of the system will enter information that is correct and truthful.

?? The knowledge base does not currently support frame-based, feature inheritance.

Hierarchical inheritance is used as the basis for substituting words in the system. That is, a word drawn from the same user-selected category is used as a replacement, as long as the word holds the same relative relationships with other substitution terms. No other "knowledge inheritance" is used within the system. In many intelligent systems, frame-based inheritance involves inheriting common features from superordinate frames or terms. (For example, if humans have heads, and John is a human, then we infer that John has a head.) This type of inheritance is not currently used in this system, although it would be apparently be desirable because it would simplify database storage. Additionally, it is believed that inheritance replicates the cognitive efficiency of human representation systems.
Project Limitations

The system is a fairly robust prototype. It amply provides proof of concept for the creation of a story/concept generator as a part of an intelligent test toolkit. This section briefly documents limits to the current system. In the first two cases these limitations are not due to theoretical limitations of computing or natural language processing, but were due to the schedule. However, the limitation concerning verb substitution is different: the feature is actually built into the system, but it does not work effectively. This exposes one weakness to the idea of using conceptual primitives, as they suggest that many verbs can be translated into a single primitive. However, the primitive does not make a good basis for text generation.

Limitations to Parsing

The User input area is moderately robust, in that it can recognize sentences with a variety of structural elements. For example, it recognizes conjunctive adverbs ("however," "consequently," etc.) at the beginning of a sentence. It is capable of determining transitive and state of being verbs. It recognizes prepositional phrases, and gerundive phrases. It recognizes the negation of any verb. It recognizes verb tense (past, present, future) and number (singular and plural). However, it does not yet recognize passive voice, nor does it recognize the object of an infinitive. Currently, it does not recognize linking verbs used with past participles (such as, has begun, was fighting), nor progressive tenses (such as, have been working) of a verb. It does not recognize compound sentences, nor complex sentences having multiple clauses. It does not recognize adjectives, although it can recognize conjunctive adverbs (such as, therefore, however).

Limitations to Verb Substitution

Substituting verbs is a particularly tricky task. The subtlety of verb choice greatly alters the meaning of sentences in most cases. Although the option for verb substitution is built-in to the system, I disabled the feature for almost every verb. Verb substitution seemed to have a distorting effect on the story/concept examples. In hindsight, common sense warns of this, yet the natural language processing literature does provide explicit guidance.

The use of conceptual primitives suggests that language (and verbs in particular) can be reduced to a small number of common forms. (Schank and Abelson (1977) suggest 11 primitives plus 5 states-of-being, although there is considerable debate.) For example, the verbs take, receive, give, hand, or present all basically mean that item X is being moved to the possession of person Y. This, therefore, should be stored in a singular representation form: the conceptual primitive PTRANS. The theory claims that of these sentences using these verbs can be represented using the same conceptual primitive. While this type of reductionist representation might be useful in building other natural language systems, it was found to be superfluous in the SCG system.

Initially the SCG system used conceptual primitives in a variety of ways as intermediary steps in language understanding. Ultimately, for this system, there was no gain in functionality in completely reducing the sentence to a primitive, and then building it back
again. In addition, primitives created a problem with verb substitution that led me to eliminate conceptual primitives almost completely from the final project design. Only a greatly simplified form of these primitives still exists within the system.

**Limitations to Semantic Networks**
The system currently allows only two distinct semantic networks to be built. In testing the system with four-five sentence story/concepts, I found that I generally did not need more than two semantic networks to handle preserving relationships. However, there is no theoretical nor programming limitation to adding additional semantic networks, should that be desirable.

**Limitations to Knowledge Base**
As discussed above, a feature inheritance could be added to the processing of the knowledge base. This would allow the system to make inferences about features for all instances of a given category. I believe that this would be a desirable feature for the knowledge base.

**Future of the Project**
The previous limitations could all be corrected with more time. They are relatively "easy fixes" that simply require more days of low-level coding.

A relatively simple yet useful spin-off of this project might entail creating a networked student interface for the system, that would enable a teacher to store the SCG output directly into a computer-based drill. The drill could be pre-programmed to provide feedback, scoring, etc.

The semantic network tool is a fully functional semantic network creation tool that can be used as the input for question-answering systems (Graesser, Byrne, & Behrens, 1992; Lehnert, 1978). This will be used as the basis for student projects in information technology courses. Additionally, one of my long-term research interests is in structural knowledge (Jonassen, Beissner, & Yacci, 1993) and the semantic network component could be modified to become a means of assessing student and faculty structural knowledge.

Ideas relating to discourse structure are relatively unexplored in this tool. It would be interesting to explore an "understanding component" to the SCG that could attempt to "understand" the story's goals with a richer internal representation. This could lead the system to create alternative discourse structures to convey that information.
**Personal Note**

I would like to thank the members of the Provost's Productivity Grant Committee for enabling me to work on this project. This was a fairly large programming effort, with many false starts and theoretical dead-ends, that consumed most of my waking hours. However, it was exciting to create this type of leading edge instructional tool and this was a fabulous opportunity to explore the practical application of research ideas in natural language processing and understanding.

I have had a paper and presentation accepted at the *AACE World Conference on Educational Hypermedia, Multimedia, and Telecommunications* in Seattle in June, 1999 and will be presenting and explaining the *Intelligent Testing Toolkit: Story/Concept Generator* system at that time. This begins to fulfil one of the project goals: to help establish the Information Technology Department at RIT as a leading edge academic force.
References


