MULTIPASS PROCESSING FOR AUTOMATIC TEXT FORECAST GENERATION

P15.9

Tracy L. Hansen *
T. J. LeFebvre
NOAA Forecast Systems Laboratory, Boulder, CO

1. INTRODUCTION

The Graphical Forecast Editor (GFE) is the grid editing component of the Interactive Forecast Preparation System (IFPS) which allows meteorologists to define weather forecasts in a digital database. Since forecasters are required to adjust and monitor literally millions of forecast points per forecast shift over the seven-day forecast period, the capability to automatically generate high-quality text from the digital data is an essential part of the system. This paper describes the challenge of generating text for a set of highly interdependent weather elements, and offers a “Multipass Model” that relies on multiple passes through a tree-like data structure representing the forecast product. This approach gives each piece of the forecast product access to the whole so that dependencies can be enforced, and then builds the text iteratively. As a simple example, the probability of precipitation wording "Probability of snow 20 percent" is dependent on the type of weather expected, i.e., "snow." So the weather phrase may be built in the first pass, and the probability of precipitation phrase can be built in a second pass having access to the results of the weather phrase. The same principle is applied to generate more complex phrasing such as local effects.

Finally, we look at the parallels to forecast methodology and the possibility of applying these concepts to the grid-editing portion of the forecast process.

2. DESCRIPTION OF PROBLEM

At first glance, the problem of representing numbers as simple phrases does not seem difficult. The phrase "Sunny. Highs in the 80's" seems as if it would be straightforward to generate given numeric values for sky cover and maximum temperatures. However, as we explore the problem further, subtleties and complexities soon arise. For example, the PoP phrase, "chance of rain..." or "chance of snow..." depends on current weather events. Also, the snow accumulation reported for today will contribute to the total snow amount reported in a later period.

As a further example, the forecast may include a chance of rain and fog in the morning, then a chance of rain in the afternoon. The system must recognize that there is a chance of rain all day and fog only in the morning.

The temporal resolution of the grids can lead to more complexity with many different weather scenarios in a 12-hr period. The system must consolidate this digital information while simultaneously producing concise yet accurate phrases. Beyond that, we need to account for sharp spatial gradients, called local effects, and report differences between areas such as "coasts" versus "inland" or "mountains" versus "valleys."

We postulate that humans do not construct their statements by sequentially adding words. Rather, they have access to and potentially use all relevant information available to them to formulate a particular phrase or sentence. We simulate this ability using a tree structure and a multipass approach.

In order to give access to each part of the whole, we create data structures in computer memory to represent the digital data and the product wording. These data structures need to be organized in an orderly fashion so that each desired piece of information can be located and accessed.

**Corresponding author address*: Tracy Hansen, NOAA Forecast Systems Laboratory, 325 Broadway R/FS6, Boulder CO 80305; e-mail: Tracy.L.Hansen@noaa.gov

3. ORGANIZING THE DIGITAL DATA – STATISTICS DICTIONARY

The digital data are sampled and analyzed to produce summary statistics such as average, minimum and maximum values. These statistics are stored in a data structure called a “dictionary,” and each piece of information can be accessed by specifying the weather element, geographical area, time range and type of statistic. For example, one can request today's maximum temperature for Boulder County Colorado.

4. ORGANIZING THE OUTPUT PRODUCT – NARRATIVE TREE

Figure 1 shows an example of a typical text product for one geographical area, perhaps a county or zone. The product consists of a series of components, each corresponding to a 12-hr time period. Each component, in turn, consists of phrases for various weather elements. Phrases may have subphrases to distinguish between time periods such as morning and afternoon.

.**TODAY...**VERY WINDY. HIGHS IN THE MID 30S. TEMPERATURES FALLING IN THE AFTERNOON. WEST WINDS 5 TO 15 MPH BECOMING NORTH 40 TO 45 MPH IN THE AFTERNOON.
.**TONIGHT...**MOSTLY CLOUDY. LOWS IN THE UPPER 20S. LIGHT WINDS.
.**WEDNESDAY...**PARTLY CLOUDY. HIGHS IN THE MID 20S. LIGHT WINDS.
.**WEDNESDAY NIGHT...**MOSTLY CLOUDY. LOW 18. A 20 PERCENT CHANCE OF SNOW.

Fig. 1. Narrative Text Product .

We can represent the product in a “tree” structure as shown in Fig. 2, which represents the hierarchy of the product components, phrases, and subphrases in the computer's memory. This representation allows examination and manipulation of the tree at all levels. Each rectangle or “node” in the tree can access and possibly change any other node prior to final output. Words for each node are stored until all processing is complete, so they are available for examination and manipulation by other nodes.

Each node of the tree can have attributes or data values associated with it. There are some standard attributes such as "words" and "geographic area." In addition, attributes can be created “on-the-fly” and stored for future reference.

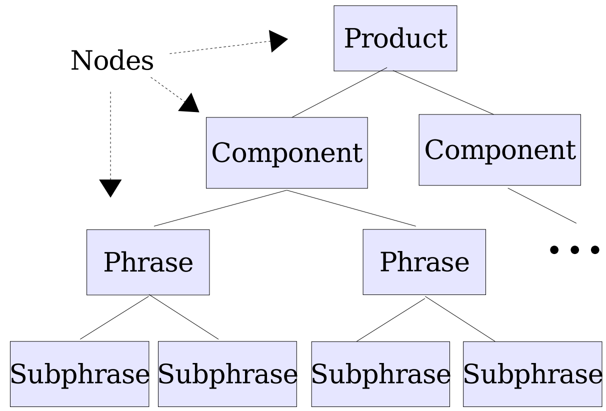

Fig. 2. Narrative Tree Structure.

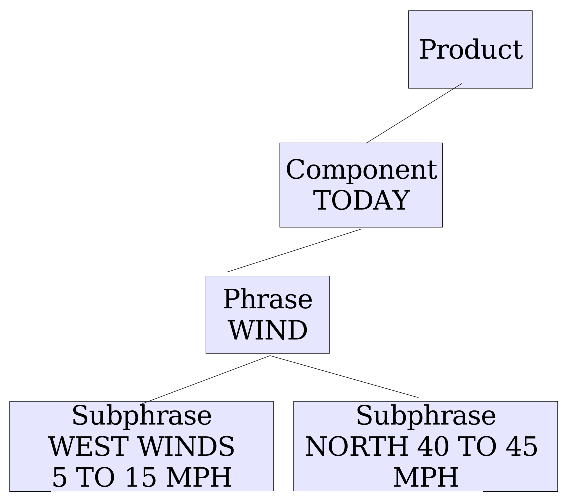

Fig. 3. Narrative Tree Example.

5. MULTIPASS PROCESSING RULES

Note that when the narrative product is structured into a tree, the order of processing is not straight-forward. There are dependencies that govern this processing order. For example, the subphrase words must be created before we can assemble the phrases, and the phrases must be assembled before we can build the component.

To address this situation in which each part is potentially dependent on the whole, we introduce the notion of “multipass” processing. Each node of the tree has a list of computer instructions or "methods" to execute. We traverse the tree from the top-down, executing the methods at each node until no changes are made to the tree. For efficiency, each method may indicate that its job is "done" and it should not be called upon again. Multipass processing introduces the potential for infinite loops. What if methods are not well-coordinated and we continue to make passes which alter the tree? For these situations, we provide a safety net by halting execution after a pass limit is reached (default limit is 20 passes).

We have some simple rules that all tree methods must follow. First, they must check for necessary data and, if it is not available yet, return without doing anything, to be called again on the next pass. If the data are available, the method does its job and returns an indication that it is "done."

For example, a method at the component level checks to see if the "words" are available for ALL of its phrases. If not, it simply returns. If so, it assembles and stores the phrases at the component level and returns an indication that it is "done" so it will not be executed again. When there has been a pass through the tree with no changes, the system stops and outputs the words from the top "product" node of the tree.

We can think of the narrative tree as a blackboard upon which we write data for other methods to examine and manipulate. Thus, we can examine the weather conditions to properly word our PoP phrase. We can sum the snow accumulation over consecutive periods to yield a total snow amount that is consistent and accurate. We can make phrases more detailed, add new phrases, or change the order of the phrases on-the-fly. No portion of the product is committed until all dependencies are accounted for and the whole is complete.

6. EXAMPLES OF THE MULTIPASS MODEL

Following are several examples to show how this model works.

6.1 *Wind Chill Wording*

Before reporting Wind Chill, we need to ensure that it is significantly less than the reported temperature. This comparison is made by the method that produces the words for the Wind Chill phrase. The method accesses the Statistics Dictionary to obtain the relevant temperature value.

6.2 *Subphrase Combining*

To avoid redundancy when subphrases would produce similar words, we must combine them into one subphrase. For example, instead of reporting "Sunny in the morning then sunny in the afternoon," we must recognize the similarity and report simply, "Sunny." In our model, we begin with a phrase node that has two subphrases, one for the morning and one for the afternoon. We set up a method for that phrase node which checks the subphrases for similarity. If found, the method replaces the two subphrase nodes with a single node that covers both the morning and afternoon time periods.

6.3 *Local Effects*

In many forecast scenarios, we find a sharp difference in data values between geographical areas. We call this a "local effect." For example, if there is a significant difference in temperature between the mountains and the plains, we want to report "Highs in the 80s except in the 60s in the mountains." To do this, we set up the node representing the temperature phrase with a method that will check for this temperature difference (see Fig. 4). If it finds one, this method will replace the original subphrase node with two: one for the mountains and one for the plains (see Fig. 5). Each of these subphrase phrase nodes then determines the words for its assigned area. In addition, the phrase node is given a special "local effect" attribute which signals it to "glue" the subphrases together using local effect wording at the end of processing.

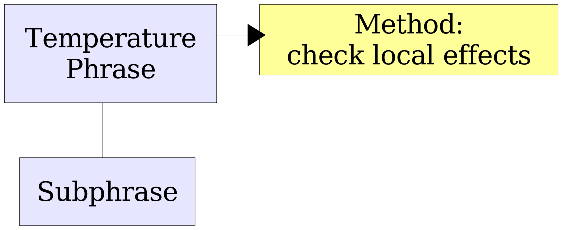

Fig. 4. Check for Local Effect.

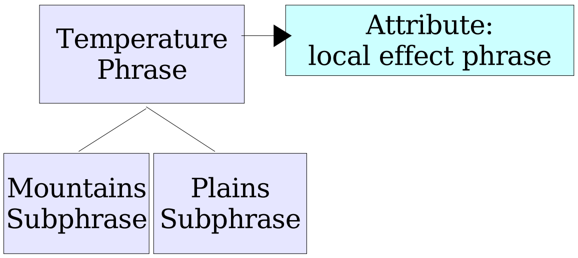

Fig. 5. Local Effect Phrase.

7. BENEFITS AND FUTURE WORK

The Multipass Model is proving to be effective as we enhance the Text Products and add new features. It is flexible while performing adequately as we make ongoing refinements to specific phrasing.

At the same time, we are realizing that we could issue the most well-formed, descriptive words, and they would be meaningless unless the underlying grids are meteorologically sound. Thus, we are focusing on the grid editing methodologies and finding parallels to our model for generating Text Products.

Because forecast weather elements are highly interdependent, the forecaster does not think linearly when creating a mental picture of the upcoming weather. The forecaster has access to and potentially uses all the information available to formulate a particular piece of the forecast puzzle.

The forecasters' tools should be compatible with their thought processes to allow them to work more efficiently. Thus, the same principles we employed for Text Products -- easy access to all relevant information and multipass processing -- will apply to grid-editing as well. We face challenges such as ensuring consistency among the many weather elements represented in the grids and creating 5-dimensional tools which must operate in space and time over multiple elements. Our experience in dealing with the complexities and dependencies inherent in text generation will serve us well as we formulate innovative processing techniques for the scientific challenges ahead.

8. REFERENCES

Hansen, T. L., T. Dankers, C. Paxton, 2003: Text Formatting with the Graphical Forecast Editor. Preprints, *19th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, Long Beach, CA, Amer. Meteor. Soc., in this preprint volume.

Hansen, T., 2000: Software approaches for the GFESuite. *FSL Forum*, N. Fullerton, Ed. Forecast Systems Laboratory, 12-15.

Hansen, T. L., M. Mathewson, M. Romberg, 2001: Forecast Methodology using GFESuite. Preprints, *17th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, Albuquerque, NM, Amer. Meteor. Soc. 42-46.

LeFebvre, T., 2000: Designing digital for better products and services. *FSL Forum*, N. Fullerton, Ed. Forecast Systems Laboratory, 24-27.

LeFebvre, T. J., Mark Mathewson, Mike Romberg and Tracy Hansen, 2001: Injecting Meteorology into the GFE Suite. Preprints, *17th Conference on Weather Analysis and Forecasting*, Albuquerque NM, Amer. Meteor. Soc. 38-41.

Nii, H. Penny, 1986, Blackboard Systems, Part One: The Blackboard Model of Problem-solving and the Evolution of Blackboard Architectures. *AI Magazine* 7 (2), 38-53.

Nii, H. Penny, 1986, Blackboard Systems, Part Two: Blackboard Application Systems, Blackboard Systems from a Knowledge Engineering Perspective. *AI Magazine* 7 (3), 82-106

Oberfield, Mark G., K. Pegion, 2002 Generating Digital Forecast Matrices From Gridded Forecasts in the Interactive Forecast Preparation System. *Interactive Symp. on the Advanced Weather Interactive Processing System (AWIPS)*, Orlando, FL, Amer. Meteor. Soc., 54-57.