

News Simulator Project

Thane Plambeck

Palo Alto, California

File Started: 1 August 2001

What is this project?

Our goal is to design and develop a system that generates random, completely fictive news stories in readable, fully grammatical English. We aim to produce stories that are indistinguishable from conventional multi-paragraph news bulletins, such as appear for example on the Associated Press wire services. We anticipate that the output of the program will be plain ASCII text (possibly XML), with HTML wrapped around it for presentation to readers using conventional web browsers.

How is it going to work?

We anticipate that the program will work from abstract *news templates* (we often simply call them *templates*) whose details are filled in via random selections from lexical, geographical, and other databases. The database sources will be entirely self-contained within the system.

A template will contain both boilerplate *fixed content* as well as abstract *adornments* that are *rewritten* to fixed content under random processes to obtain the final “news” story output.

For example: consider the following initial portion of an actual news story, taken from Reuters wire services on 28 November 2000:

Six dead in Canadian bus, truck crash

Tuesday, November 28, 2000 12:02pm

REVELSTOKE, British Columbia, Nov 28 (Reuters) - Six people were killed and 21 injured when a bus filled with Taiwanese tourists collided with a truck in a tunnel in the mountains of British Columbia, police and media reports said on Tuesday.

The cause of the accident on the Trans-Canada Highway on Monday about 50 kilometers (30 miles) east of Revelstoke near Rogers Pass was not immediately known.

This news story can be changed into a different one by simple local changes to it. For example, the number of dead might be changed to 10; the tourists could be identified as Brazilian; the date moved three years earlier; the location could be changed; etc. In fact, the whole article is stereotypic of similar motor vehicle accident accounts that have been described for many years on the news wires.

Here is a corresponding simple news template that illustrates the basic idea and our associated terminology:

```

$[ORDINAL(x)]$ dead in Canadian bus, truck crash

$[DATE(y)]$ $[TIMEOFDAY]$

REVELSTOKE, British Columbia, $[MONTHDAY(y)]$ (Reuters) -
$[(x)]$ people were killed and 21 injured when a bus filled with $[NATIONALITY]$
tourists collided with a truck in a tunnel in the mountains of British Columbia,
police and media reports said on Tuesday.

The cause of the accident on the Trans-Canada Highway on Monday
about 50 kilometers (30 miles) east of Revelstoke near Rogers
Pass was not immediately known.

"It stretched our medical personnel to the max," Mountie Gerry
Komax told reporters.
```

The notation `$[...]$` is used to start and end each adornment. This notation separates an adornment from the fixed content that surrounds it. The keywords `ORDINAL`, `DATE`, `TIMEOFDAY`, `MONTHDAY`, and `NATIONALITY` are examples of particular *adornment types*. (See the **Glossary** for a current list of adornment types).

A specific adornment type may or may not require *variables*. For example, `TIMEOFDAY` and `NATIONALITY` have no variables, while `ORDINAL` has one variable, named “x.” In reading a template, the first occurrence of a particular variable globally within the template is said to be *free*, while all subsequent appearances of that variable are said to be *bound*. In the example above, the “y” in

```
$[DATE(y)]$
```

is free; while the subsequent “y” in

```
$[MONTHDAY(y)]$
```

is bound.

The process of *rewriting* a template to fixed content is one of determining random assignments for the free variables and their adornments, while respecting the bindings established for the earlier adornments and free variables in the rewriting of in the subsequent content.

Generating High Quality Stories

In constructing variants of articles using mechanisms such as are described above, *specificity* and *variability* play important roles in determining the credibility of the results. For example, it wouldn’t do to generate an endless sequence of stories, all reported from Revelstoke on the Trans-Canada highway, and commented upon by Mountie Gerry Max. Therefore, we propose to draw from real world descriptive databases of places, highways, cities, professions, person names, etc in generating article variants.

Where are the files located?

Main work for this project (on Thane's home machine):

c:/work/newsSimulator

Tools

We anticipate developing on a Windows platform using Unix-style tools. Here are some of the tools we'll probably be using to develop on Windows:

cygwin bash

byacc

flex

cpp

gcc

Glossary**References**