SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator

Somayajulu G. Sripada and Ehud Reiter
Department of Computing Science
University of Aberdeen
{ssripada,ereiter}@csd.abdn.ac.uk

Ian Davy
Weathernews (UK) Ltd.
Aberdeen
iand@wni.com

Abstract

Numerical weather prediction (NWP) models produce time series data of basic weather parameters which human forecasters use as guidance while writing textual forecasts. Our studies of humans writing textual weather forecasts led us to build **SUMTIME-MOUSAM**, a text generator that produces textual marine weather forecasts for offshore oilrig applications. **SUMTIME-MOUSAM** separates control and processing. As a result of this forecasters can tailor the output text using control data derived from end user profiles. In this paper we describe the design and the implementation details of **SUMTIME-MOUSAM** which is currently being used by our industrial collaborator. Output from our system is post-edited by forecasters before communicating it to the end-users. We also briefly describe an evaluation of our system using the post-edit data.

1 Introduction

Almost all modern weather forecasting is done using guidance from Numerical Weather Prediction (NWP) models; time series data from the NWP models is used to produce required forecast products. There is increasing interest in tools to produce forecast texts from NWP forecast data such as FoG (Goldberg et. al., 1994) and ICWF (Ruth and Peroutka, 1993). In this paper we describe **SUMTIME-MOUSAM,** a text generator that produces textual marine weather forecasts for offshore oilrig applications. Forecasters can control the processing in our system by changing parameters in an external control data file. Using this mechanism, forecasts can be tailored to the specific end user requirements. Our industrial collaborator, Weathernews (UK) Ltd. is currently using the system to generate an initial version, which is then edited by human forecasters before communicating to the end-users. We have used the post-edit data to carry out an evaluation of **SUMTIME-MOUSAM**, which is briefly described in the paper.

2 Background

SUMTIME is an ongoing research project aiming to develop generic techniques to produce textual summaries of time series data (Sripada et. al., 2001). In order to achieve our objective we study summarization of time series data in three domains, meteorology, gas turbines and neonatal intensive care unit (NICU). In the domain of gas turbines we are working on summarizing sensor data from an operational gas turbine (Yu et. al., 2003) for the maintenance engineers. In the domain of neonatal intensive care we are working on summarizing physiological data to the doctors in the neonatal ICU (Sripada et. al., 2003a). In the domain of meteorology we are working very closely with a weather forecasting company, Weathernews (UK) Ltd., Aberdeen to understand the process involved in producing textual forecasts from NWP model data.

3 SumTime-Mousam

In **SUMTIME**, we have carried out a variety of knowledge acquisition (KA) activities using multiple techniques developed in the expert system community to understand how humans perform weather forecasting (Reiter, Sripada and Robertson, 2003). Among other things our KA has highlighted the importance of:

3.1 Sensitivity to End User

One key observation was that content in the forecast depends upon the end user. Different oilrigs need different pieces of information in a weather forecast. An offshore oilrig in the North Sea might have different requirements from one in the Persian-gulf because of the differences in their structural designs.

3.2 Forecaster Control

It is important that the text generator is configurable by the forecaster to adjust the output without writing new code. The need for this might arise due to changes in end user requirements. Forecasters understand the end user requirements and also understand how they affect the generation of forecast texts. In this sense, forecasters are experts of domain communication knowledge (DCK) as described in (Kittredge and Polguere, 2000).

3.3 Architecture

Based on the observations from the KA we built **SUMTIME-MOUSAM**, a system that generates marine forecasts. This section describes the details of the system. Our system follows the pipeline architecture for text generation (Reiter and Dale, 2000) as shown in Figure 1. The rest of this section describes each of the components in detail explaining how weather data is converted stage wise to text.

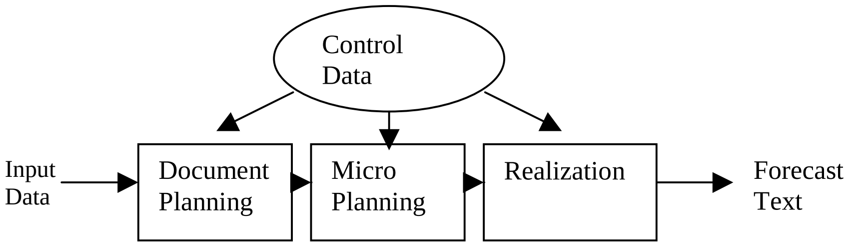

Figure 1 Architecture of **SUMTIME-MOUSAM**

3.4 Input

Input to **SUMTIME-MOUSAM** is obtained by sampling forecaster edited data from the NWP model prediction at the required grid point. Table 1 shows a portion of input to our system. Full data set includes approximately forty basic weather parameters. Input data can be described as a multivariate time series. The time interval is 3 hours and data predictions extend unto 72 hours (3 days) from the issue time of the forecast. In table 1 we showed only the first day forecast.

Table 1. Weather Data produced by an NWP model for 12-Jun 2002

Date and Time	Wind Dir	Wind Speed 10m	Wind Speed 50m	Gust 10m	Gust 50m
12-06-2002 03:00:00	SW	9.0	11.0	11.0	14.0
12-06-2002 06:00:00	W	10.0	12.0	12.0	16.0
12-06-2002 09:00:00	W	11.0	14.0	14.0	17.0
12-06-2002 12:00:00	WSW	10.0	12.0	12.0	16.0
12-06-2002 15:00:00	SW	7.0	9.0	9.0	11.0
12-06-2002 18:00:00	SSW	8.0	10.0	10.0	12.0
12-06-2002 21:00:00	S	9.0	11.0	11.0	14.0
13-06-2002 00:00:00	S	12.0	15.0	15.0	19.0

3.5 Output

Figure 2 shows the first day forecast text generated by our system. Output forecast text is organized into various forecast elements such as Wind, Wave, Weather etc. Each of the forecast elements describes a few basic weather parameters. For example, the wind part of the forecast has been generated using the data shown in Table 1. Some of the data channels are considered primary while others secondary based on their contribution to a forecast element. For example wind direction, and wind speeds at 10m and 50m are primary channels while gusts at 10m and 50m are secondary. Primary channels and secondary channels are processed differently while producing forecast text.

3.6 Control Data

One of the main features of **SUMTIME-MOUSAM** has been that its output-forecast text can be varied depending on the end user preferences. These preferences are stored in external files as control data tables. Using these tables forecasters can alter the output in terms of its content (level of detail) and also its style. More details on user modeling **SUMTIME-MOUSAM** have been described in (Reiter, Sripada and Williams, 2003). We describe the use of control data as they are used in the three stages of processing.

---

2. FORECAST 6 - 24 GMT, Wed 12-Jun 2002

WIND(KTS)

10M: W 8-13 backing SW by mid afternoon and S 10-15 by midnight.

50M: W 10-15 backing SW by mid afternoon and S 13-18 by midnight.

WAVES(M)

SIG HT: 0.5-1.0 mainly SW swell.

MAX HT: 1.0-1.5 mainly SW swell falling 1.0 or less mainly SSW swell by afternoon, then rising 1.0-1.5 by midnight.

PER(SEC)

WAVE PERIOD: Wind wave 2-4 mainly 6 second SW swell.

WINDWAVE PERIOD: 2-4.

SWELL PERIOD: 5-7.

WEATHER: Mainly cloudy with light rain showers becoming overcast around midnight.

VIS(NM): Greater than 10.

AIR TEMP(C): 8-10 rising 9-11 around midnight.

CLOUD(OKTAS/FT): 4-6 ST/SC 400-600 lifting 6-8 ST/SC 700-900 around midnight.

Figure 2. Forecast Text Produced by **SUMTIMEMOUSAM** for the AM of 12-Jun 2002. The Wind part of the forecast has been generated from the data, shown in Table 1

3.7 Document Planning

Document planning is responsible for selecting the 'important' data points from the input data and to organize them into a paragraph. The important question in summarization is 'what data points from the input should be included in the summary?' Any model of data summarization needs to find ways to reduce the size of the input data set (or improve its accessibility) without significantly altering its content (or informativeness). This process is sensitive to the domain constraints such as limits on parameter values. It is clear from our own studies on data summarization and also from the earlier studies by others (Goldberg et. al. 1994, Boyd, 1998; Kulkich, 1983) that data summarization needs data analysis to determine the trends and patterns present in the input data set.

The data analysis we perform on the input data is what is known as segmentation. Segmentation is the process of fitting linear segments to an input data series keeping the maximum error introduced in segments to be lower than the user defined value. There are many algorithms for segmentation developed in the KDD community. These algorithms differ from each other in the control information they use and the way they process data (such as top-down and bottom-up). We have selected one of them known as the bottom-up algorithm (Keogh, 2000). We have modified this algorithm, which is primarily suited for data mining, in a number of ways as described in (Sripada et al, 2003b) to make it suitable for use in a communication application such as producing forecasts for specific end users. Most of these modifications use control data to produce segments that fulfill end user preferences.

For the example data shown in Table 1, segmentation of wind speed at 10m height produces one segment (one line joining 10.0 knots at 0600 to 12.0 knots at 2400) and segmentation of wind direction produces two segments, (one line joining W at 0600 to SW at 1500 and a second line joining SW at 1500 to S at 2400). For more details of our segmentation algorithm, please refer to (Sripada et. al, 2002).

The output of data analysis is converted into a conceptual representation. Conceptual representation of wind forecast consists of a tuple with the following elements

- Time
- Wind speed lower range (lower bound on predicted wind speed computed using control data)
- Wind speed upper range (upper bound on predicted wind speed computed using control data)
- Wind direction (in degrees)
- Modifiers (gusts, shower, steady-change, gradual-change)

In our example case, we have two segments of wind direction and one segment of wind 10m. Based on our corpus analysis we simply take the union of all the segments. Therefore we have three tuples

(0600, 8, 13, W, nil) (1500, 8, 13, SW, nil) (2400, 10, 15, S, nil)

The lower and upper range values of wind speed are computed using data from the control tables.

3.8 Micro-planning

Micro planning is responsible for lexical selection and ellipsis. It receives a sequence of tuples from the document planner and performs two tasks on these tuples. Both the tasks have been implemented using the rules obtained from corpus analysis and other KA tasks (Reiter and Sripada, 2002; Reiter and Sripada, 2003). The first task of the micro-planner is to make lexical choices. For this purpose, each forecast text element is divided into a sequence of higher level phrases and connectives. Initially, we create a high level phrase corresponding to each input tuple. For example Wind10m text is generated by a sequence of wind phrases. At a lower level each wind phrase is in turn divided into direction phrase, speed phrase, verb phrase and time phrase. Each of these phrases is modeled after the PSLex-CaseFrame as described in (Reiter and Dale, 2000).

Table 2. Lexicalized Phrases with Ellipsis Marking

Tuple	Wind Phrase Components	Flags
(0600, 8, 13, W, nil)	**Dir phrase 1:** W; **Speed Phrase 1:** 8-13; **Time Phrase 1:** by early morning	Time phrase is elided
(1500, 8, 13, SW, nil)	**Dir phrase 2:** SW; **Speed phrase 2:** 8-13; **Time Phrase 2:** by mid afternoon; **Verb Phrase**: backing	Speed phrase is elided
(2400, 10, 15, S, nil)	**Dir phrase 3:** S; **Speed Phrase 3:** 10-15; **Time Phrase 3:** by midnight; **Verb Phrase**: backing	Verb phrase is elided

For our example tuples created in the previous section, we have lexicalised phrases as shown in the middle column of Table 2. The second step is to look for cases of ellipsis. This is once again guided by rules extracted from corpus analysis and other KA tasks. A few sample rules at this stage are:

- Always suppress the time phrase for the first wind phrase (because this is the beginning of the forecast)

- Suppress direction phrase if same as previous direction phrase
- Suppress speed phrase if same as the previous speed phrase
- Suppress the entire wind phrase if both speed phrase and direction phrase are suppressed

During this step flags are set to mark ellipsis information. For our example case the flags are marked as shown in the last column of Table 2.

3.9 Realization

Lexicalised phrases marked for ellipsis created in the previous stage are input to this stage. Realization is essentially responsible for ordering of the phrases in the output and also to perform punctuation tasks. For our example case we have the final output for Wind10m:

W 8-13 backing SW by mid afternoon and S 10-15 by midnight.

4 Evaluation

Weathernews has been using our system for the past one year running it to generate an initial version of the forecast. Forecasters then edit the initial version to produce the final forecast that is communicated to the end-users. We measured the number of edits forecasters make as a metric for evaluating our system. More precisely, Weathernews currently use **SUMTIME-MOUSAM** as shown in Figure 3. The raw NWP data is first edited by Weathernews staff using their internal Editing Tool, based on their specialized meteorological knowledge. **SUMTIME-MOUSAM** is then run on the output of the Editing Tool (Data 1 in Figure 3), in order to describe the edited data to the forecaster himself. This produces Forecast Text 1 in Figure 3. The forecaster then adjusts the data (not the text) in cases where he believes the NWP prediction is incorrect; this produces modified data (Data 2). **SUMTIME-MOUSAM** is then run again on the modified data, to produce a draft forecast for the customer; this is Forecast Text 2. The forecaster then manually edits this text, and produces the final Forecast Text 3.

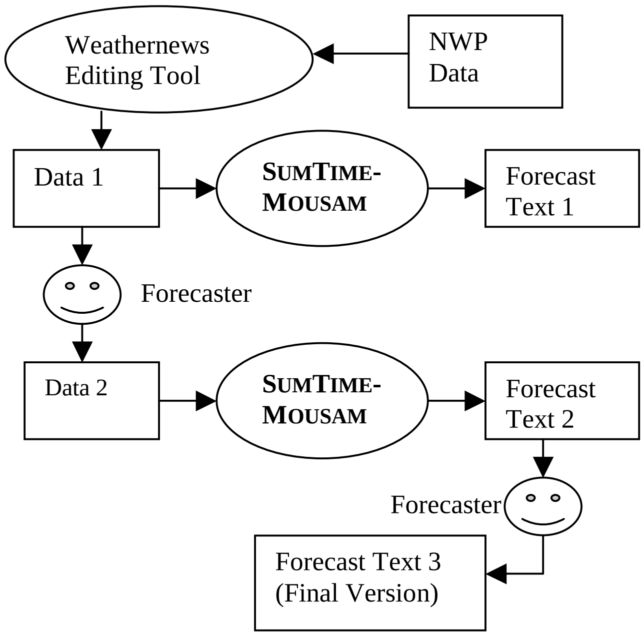

Figure 3. Testing Scheme used in Industrial Evaluation

Our evaluation measures the edits made by the forecaster in the final step, when he produces Forecast Text 3 from Forecast Text 2. This measures how many problems need to be fixed in the generated texts before they are fit to be used to communicate information effectively for humans.

In order to carry out the evaluation, we have first divided the forecast texts into phrases and aligned the phrases produced by our system (forecast text 2) and those from the edited forecast (forecast text

3). Initially we have carried out this process manually (Sripada et al 2003b). Later we have developed software to carry out this procedure automatically.

In our evaluation, not all phrases could be aligned; we failed to align 17% of phrases, which are mainly due to the difference in segmenting input data by our system and the forecasters. The successfully aligned phrases have been then compared for exact matches. 43% of total phrases match word to word. That is human forecasters did not feel the need to edit these phrases at all. The remaining phrases have been edited by forecasters, which resulted in 40% of mismatching phrases. These edits are being studied currently to make changes to **SUMTIME-MOUSAM**.

5 Conclusion

In this paper we have described a data to text weather forecast generator that allows forecasters to make adjustments to the output text to meet the specific needs of end users. Forecasters through their experience understand end user requirements and translate them to control data driving the data analysis techniques that determine the content of the forecast.

References

1. S. Boyd. 1998. TREND: a system for generating intelligent descriptions of time series data. In *Proceedings of the IEEE International Conference on Intelligent Processing Systems (ICIPS-1998).*
2. E. Goldberg, N. Driedger, and R. Kittredge, 1994. Using Natural-Language Processing to Produce Weather Reports. IEEE Expert; 9:45-53.
3. J. Yu, J. Hunter, E. Reiter and S. Sripada, 2003. SumTime-Turbine: A Knowledge-Based System to Communicate Time Series Data in the Gas Turbine Domain. To appear in The 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Loughborough, UK, June 23-26, 2003.
4. E. Keogh, S. Chu, D. Hart and M. Pazzani, 2001. An Online Algorithm for Segmenting Time Series. In: Proceedings of IEEE International Conference on Data Mining, , pp 289-296.
5. R. I. Kittredge and A. Polguere, 2000. The Generation of Reports from Databases. In: Dale, R., Moisl, H. and Somers, H. (ed) Handbook of Natural Language Processing, pp. 261-304.
6. K. Kukich, 1983. Design and implementation of a knowledge-based report generator. In: Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (ACL-1983), pp 145-150.
7. E. Reiter and R. Dale, 2000. Building Natural Language Generation Systems. Cambridge University Press.
8. E. Reiter and S. Sripada, 2002. Human Variation and Lexical Choice. Computational Linguistics 28:545-553, 2002.
9. E Reiter and S Sripada, 2003. Learning the Meaning and Usage of Time Phrases from a Parallel Text-Data Corpus. In Proceedings of the HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data, pages 78-85.
10. E. Reiter, S. Sripada and S. Williams, 2003. Acquiring and Using Limited User Models in NLG. In Proceedings of ENLGW 2003, pp 87-94, Budapest, Hungary.
11. E. Reiter , S. Sripada, and R. Robertson, 2003. Acquiring Correct Knowledge for Natural Language Generation. Journal of Artificial Intelligence Research, 18: 491-516, 2003.
12. D. P. Ruth, and M.R. Peroutka, 1993: The Interactive Computer Worded Forecast. Preprints 9th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Anaheim, Amer. Meteor. Soc., 321-326.
13. S. Sripada, E. Reiter, J. Hunter and J. Yu, 2001. Modelling the task of Summarising Time Series Data using KA Techniques. In: Macintosh, A., Moulton, M. and Preece, A. (ed) Proceedings of ES-2001, pp 183 –196.
14. S. Sripada, E. Reiter, J. Hunter and J. Yu, 2002. Segmenting Time Series for Weather Forecasting. In: Macintosh, A., Ellis, R. and Coenen, F. (ed) Proceedings of ES2002, pp. 193-206.
15. S. Sripada, E. Reiter, J. Hunter and J. Yu, 2003a. Summarizing Neonatal Time Series data, Proceedings of the Research Note Sessions of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), pp. 167-170, Budapest, Hungary, 2003.
16. S. Sripada, E. Reiter, J. Hunter and J. Yu, 2003b. Generating English Summaries of Time Series Data using the Gricean Maxims. In Proceedings of KDD-2003, pp 187 –196.