The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Content Selection for Time Series Summarization Using Case-Based Reasoning

Neha Dubey, Sutanu Chakraborti, Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Chennai 600 036, India
{nehamay, sutanuc}@cse.iitm.ac.in, khemani@iitm.ac.in

Abstract

We propose a Case-Based Reasoning(CBR) approach for content selection, which is an intermediate step towards generating textual summaries of time series data in the weather prediction domain. Specifically, we handle two significant challenges, the first involving multivariate data that warrants modeling of the interaction of two 'channels' (wind speed and direction in our context) and the second involving the effective integration of domain-specific knowledge in the form of rules with data from a case library of past instances of content selection. We present an approach that uses domain knowledge to transform a given raw time series instance into a representation that facilitates effective retrieval of relevant cases, which are then used for change point prediction. We empirically demonstrate that our approach combining CBR and domain rules outperforms classical content selection mechanisms that are based on rules or heuristics alone as well as those that are purely data-driven.

Introduction

Data-to-text generation is a subfield of Natural Language Generation (NLG) in which text is generated from non-linguistic data sources such as time series and sensor logs. *Content Selection* is an integral part of Data-to-text systems. It deals with the task of choosing the relevant information out of input data that needs to be represented in the generated text. For example, in the domain of weather prediction, a weather forecaster decides changes in the wind speed and direction that are important at a particular time of the day (Sripada et al. 2002a), whereas to generate feedback report for a student, factors like the hour of studies, health, attendance, etc. are important.

In this work, we propose a Case-Based Reasoning (CBR) (Kolodner 1992) approach for content selection and demonstrate its use in weather domain. Content selection in time series summarization involves selecting few representative change points from time series that capture the trend information. These representative points are used to generate the textual summary of time series.

In the past, several NLG systems have been built to generate text mainly in the weather domain. In knowledge rich systems, content selection is done from the top down knowledge acquired from experts (Sripada et al. 2002a); in contrast, knowledge-light systems use bottom-up knowledge acquired from the corpus (Belz 2008). While the former class of systems are hard to build because of knowledge acquisition bottleneck, knowledge-light systems need a large, correct and consistent corpus.

In this paper, we propose to strike a middle ground between top-down and purely data-driven extremes by using a mix of rules and learning from the corpus. Towards this end, we purpose a CBR system that uses few specific rules typically used by experts for content selection from time series data. CBR works by recalling past experiences and is based on the premise that similar problems tend to recur and have similar solutions. In the current context, we have a case library consisting of cases, with each case encoding a representation of a time series as its problem component, and a representation of content useful for mapping the time series to text as its solution component. An incoming time series is matched against problem components of stored cases, and the solutions of similar cases are used to decide the parameters that govern content selection. More specifically, in this work we focus on the following challenges :

- Arriving at a similarity measure for multivariate time series is hard because interaction between attributes (channels) like wind speed and wind direction needs to be modeled. It is not enough to individually analyze univariate time series corresponding to each channel.
- Since a domain expert often has different interpretations for two time series that look very similar to a non-expert's eyes, more external knowledge from domain is required to get a better representation of a given time series.

Related Work

The earlier approach (Sripada et al. 2002a) for content selection from weather time series uses external knowledge from experts to summarize a time series. This knowledge, in the form of error thresholds, is used to control the level of summarization. Although the error thresholds make the system configurable to end user, it is difficult to get these thresholds in the absence of experts.

To overcome the knowledge acquisition bottleneck in the above approach, a purely data-driven approach was pro-

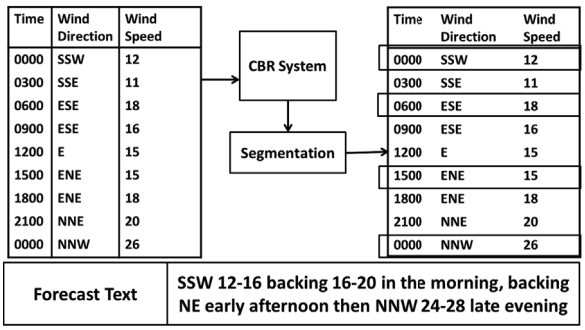

Figure 1: System architecture

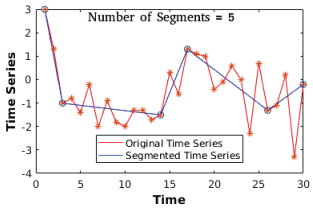

Figure 2: Segmentation example

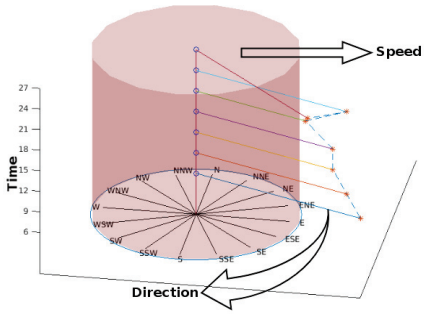

Figure 3: Wind time series

posed by Sowdaboina, Chakraborti, and Sripada (2014), which uses Machine Learning techniques to induce a model for identifying the change points in a time series based on some handcrafted features extracted from the time series. However, the approach only considers points from one channel, i.e., wind speed, and fails to handle the cases where the text summarizes the other channel or both channels (wind speed and direction) together.

There are other weather forecast text generation systems - CBR (Adeyanju 2012) and probabilistic context-free grammar (PCFG) based systems(Belz 2008). Since these systems use the time series reverse engineered from the text, the focus is on other NLG tasks like microplanning and realization and not on content selection.

Our System

Our system has two components: the first decides on the required abstraction to summarize a given time series, and the second component generates representative points from a time series to report in the text. Thus, the input to the system is a time series of wind speed-direction and the output is speed-direction tuples at chosen time intervals. Our system's architecture is shown in Figure 1.

Content generation from time series involves striking a trade-off between minimizing the number of change points reported and maximizing the faithfulness of its approximation. The error in approximation can be reduced by increasing the length of the summary. In the proposed work, we predict the number of change points using CBR, and then generate an approximation to the time series that minimizes the approximation error.

To approximate a given time series, we use the optimal segmentation algorithm (Bellman 1961). Segmentation is the process of approximating a time series with straight line segments. For example, Figure 2 shows the segmentation of a time series with 5 segments. Given the number of segments, the optimal segmentation algorithm globally minimizes the error of approximation.

In weather domain, the raw wind time series is taken as the starting point for content selection. The output is a set of selected wind states. For example, the text in Figure 1 has 4 wind states , viz. (SSW, 12-16), (ESE,16-20), (NE,-) and (NNW 24-28). The count of wind states is used as input to the segmentation algorithm. In our case, the number of wind states is the same as the number of segments.

The task of predicting the number of segments is formulated as a CBR regression problem, with the number of segments in a time series as the solution components of the cases. Once the number of segments is predicted using CBR, we segment the query time series to generate change points using the estimated segment count. We will discuss the full procedure in the following sections.

Predicting the number of segments using CBR

In weather domain, days with similar weather conditions have the similar forecast text, based on this observation, we assume the CBR hypothesis that *similar wind time series have similar number of wind states in forecast text*. Each case has a representation of the time series as its problem side and its corresponding number of segments as the solution side. The representation of the time series is wind vector representation as shown in Figure 3.

Transformation of a time series When experts look at a time series, they implicitly transform it using their domain expertise to yield a view that facilitates text generation that satisfies the intended communication goal. Therefore, our representation of cases should ideally capture this domain knowledge. We use some basic observations gathered by Sripada et al. (2002a) to form rules that transform the time series. The rules used in our cases are: 1. If the speed is fluctuating within a small range τ_1 and it has a flat trend for the full day, then the changes in direction are less important than speed. 2. If the direction is constant during the day, then the overall trend of speed is important (minor fluctuations during the day are ignored). 3. If the wind is strong breeze or

high wind[1] during the day and the direction changes more than half times a day, then direction changes are more important than speed changes.

To incorporate these rules in our case, we use two weights w_1 and w_2 to weigh the changes in speed and direction, respectively. The change in speed and direction at time t can be defined as follows:

Let $slope\_speed_t$ be the slope of speed at time t, hence, $slope\_speed_t = (speed_t - speed_{t-1})/(t_t - t_{t-1})$; Let $slope\_direction_t$ be the slope of direction at time t, hence, $slope\_direction_t = (direction_t - direction_{t-1})/(t_t - t_{t-1})$; Then, the angle of change at time t for speed can be defined as $\theta\_speed_t = \arctan(slope\_speed_t)$, and for direction, $\theta\_direction_t = \arctan(slope\_direction_t)$.

Now, according to our rules, the weights have following inequalities:

1. For rule 1, $w_1 > w_2$ and $w_2 < 1$.
2. For rule 2, $w_1 < 1$.
3. For rule 3, $w_2 > w_1$.

These weights and thresholds are set by using cross-validation. Once we have the weight values, we can change our θ_speed_t and $\theta\_direction_t$ by multiplying them with respective weights. This results in $\theta\_speed_t\_new$ and $\theta\_direction_t\_new$. Now, we have the change values at every time t, new time series can be formed using the following algorithm.

Algorithm 1: Transformed Time Series

Result: $T_{new} \leftarrow ((speed\_new_1), (direction\_new_1))$,... ...$((speed\_new_i), (direction\_new_i)..)$; where $i \in 1, 2, 3, ...n$

$speed\_new_1 \leftarrow speed_1$; $direction\_new_1 \leftarrow direction_1$

for $i \leftarrow 2$ **to** n **do**

$speed\_new_i \leftarrow speed\_new_{i-1} + \tan(\theta\_speed_{t_i}\_new) * (t_i - t_{i-1})$

$direction\_new_i \leftarrow direction\_new_{i-1} + \tan(\theta\_direction_{t_i}\_new) * (t_i - t_{i-1})$

where i is the time index in time series and t_i denotes time i

Retrieval To retrieve the most similar time series for a new time series, we use dynamic time warping (DTW) (Sakoe and Chiba 1978) on wind vector time series. DTW aligns two time series by scaling/shrinking on time axis. The DTW distance between two time series T_1 and T_2 is

$$DTW(i,j) = \min\{DTW(i-1,j), DTW(i,j-1), DTW(i-1,j-1)\} + vectordist(i,j)$$

where $DTW(1,1)$=$vectordist(1,1)$, which is the distance between first points of both time series, and i and j are the time indices in time series T_1 and T_2, respectively. The distance measure *vectordist* in our case can be defined as

$$vectordist(i,j) = \sqrt{s_{T_{1i}}^2 + s_{T_{2j}}^2 - 2 * s_{T_{1i}} * s_{T_{2j}} * cos(d_{T_{1i}} - d_{T_{2j}})}$$

where $s_{T_{1i}}$, $d_{T_{1i}}$ denote speed and direction at time i of time series T_1 and similarly $s_{T_{2j}}$, $d_{T_{2j}}$ is defined for time series T_2. The total distance between two time series is $dist_{DTW}(T_1, T_2) = DTW(n, n)$, where n is the number of points in time series.

[1] www.metoffice.gov.uk/guide/weather/marine/beaufort-scale

Selection of points

The algorithm (Bellman 1961) takes as input number of segments as predicted by the CBR system, and outputs an approximated (segmented) time series, which has the least possible error of approximation. The points to report in the text are the endpoints of each segment in the time series.

Experiments

We used the SUMTIME-MAUSAM parallel corpus (Sripada et al. 2002b) of 1045 numerical weather data and human written forecasts. We take wind time series where speed and direction are predicted using NWP(Numerical Weather Prediction) model in each 3-hour interval. All results mentioned are obtained by performing 5-fold cross-validation with 80-20 split.

Ground truth We construct our ground truth by using the corpus (Sripada et al. 2002b), which has text aligned with the corresponding entry in numerical time series. We take only those time series-text pairs, in which all phrases in the text are aligned with some entry in time series. As humans tend to use interpolation for segmenting a time series(Sripada et al. 2002a), we construct a time series from the text by interpolating between two consecutive values of time series. For example, given a matched time as (6:00, 12:00, 24:00), and points in forecast text as (34-38, SSW), (26-30, -), (22-26, -); we reconstruct the time series as (36.0,SSW), (32,SSW), (28,SSW), (27,SSW), (26,SSW), (25,SSW), (24,SSW).

Experiment Design We have designed the experiment in two parts:

1. We predict the number of segments of a time series and the evaluation measures used are accuracy for classification and mean squared error(MSE) for regression (i.e., K_{error}) .
2. We segment the time series with predicted segment count and call the segmented time series as $P_{timeseries}$. The error between ground truth time series G_t and segmented time series $P_{timeseries}$ (i.e., $Error(P_t, G_t)$) is calculated.

Methods Compared

To the best of our knowledge, there is no existing work that can be directly compared with our work in terms of content selection effectiveness. While Sripada et al. (2002a) evaluated using post edit data, Sowdaboina, Chakraborti, and Sripada (2014) used segment count for speed channel alone to detect change points in the time series. Therefore, we compare our work with following methods:

Elbow method: We plot the error of approximation against varying number of segments for a time series. At some point, the marginal error reduction will drop significantly, this elbow point is chosen as the number of segments for a given time series.

Decision tree: To predict segment count, we take segment count as class label. Features extracted from the time series include minimum, maximum, range, end to end slope, regression error, standard deviation of speed and direction, respectively. These statistical features capture the variation in time series and helps in determining the number of segment. For example, a high mean of wind speed and a high standard deviation of direction tends to have a large segment count.

CBR system: We retrieve similar time series by using a similarity measure based on DTW and estimate the segment count. Next, we segment the time series using the estimated segment count and report the end points of segments.

CBR system+rules: Finally, we apply domain specific rules to transform our time series and then use DTW similarity to retrieve similar time series to predict the segment count. The time series is segmented with the estimated count and the end point of segments are reported in text.

Results and Discussion

The results of all methods are summarized in Table 1. Analysis of the CBR system reveals that in some cases, it is unable to capture the domain knowledge appropriately. For example, if the direction changes only at night, i.e., 24:00, it may get ignored. However, this can be matched with other time series in which direction changes at evening, i.e., 21:00 by stretching along the time axis. In the example above, time scaling becomes undesirable for the domain. Thus, we need more knowledge than just shape matching to cover these domain specific cases. This knowledge can be captured either in the similarity measure or by using better representation of time series.

We exploit the above fact and try to capture the relative importance of speed and direction by using some rules. However, we could apply more rules for better representation if we had access to experts knowledge.

We analyzed the misclassified cases (classification) for the number of segments, which are around 38% in our case, and discovered that more than half (22%) of misclassified cases are consistently misclassified in all of the above methods. We suspect these cases need extra domain knowledge. We also observed that, given a raw time series, it is often difficult even for a human to decide the exact segment count. This suggests that a forecaster is either using her expertise or some other domain specific tacit knowledge to decide upon the segment count.

As evaluation in NLG is hard, specifically when it is difficult to choose between several competing candidates that summarize the time series equally well, it may be misleading to conclude readily that the approach failed in the 38% cases that are incorrectly predicted. For example, even a forecaster may find it hard to determine whether a time series should have 2 or 3 segments. Therefore, the content selection algorithm is useful even when number of segments is not identical to that in the ground truth, but close to it, as long as the time series P_t and G_t have low error difference. Since our ground truth G_t is constructed using the time matching information from (Sripada et al. 2002b), which involves manually reverse engineering a time series from text, there is an element of subjectivity in choosing G_t from several possible candidates, hence we can not overtly rely on approximation error either. Therefore we evaluate using both errors measures K_{error} and $Error$ and give more importance to K_{error} than $Error$.

The results clearly illustrate that using a judicious mix of cases and rules holds promise for content selection. Further, the issue of handling interaction between channels (attributes) has been handled in the process of creating richer case representations. This also has a cognitive appeal in that it comes closer in modeling how experts would combine their general domain knowledge with their specific experiences in generating textual summaries from time series to arrive at a representation of content, that can be effectively mapped to text.

Approach	Accuracy(%)	K_{error}	$Error(G_t, P_t)$
Elbow method	30.49	-	3.05
Decision Tree	49.87	-	2.53
CBR	57.87	0.40	2.876
CBR+Rules	61.96	0.37	2.876

Table 1: Accuracy of number of segments(Classification); K_{error} : MSE between actual and estimated number of segments (Regression); $Error$: Error between ground truth G_t and segmented input time series P_t

Conclusion and Future Work

In this paper, we presented a CBR system for choosing content to generate weather forecast texts. First, the system uses few domain specific rules to transform time series. Next, the number of representative points for a new time series are estimated using similar time series stored in the case base. The new time series is segmented using the estimated number of points such that the error of segmentation is minimal. The endpoints of segments are reported to generate text. We have empirically demonstrated the effectiveness of this approach over purely data driven as well as purely top down systems, and also addressed the issue of handling interaction between channels. For the purpose of final evaluation of generated text, we plan to extend our work to make an end-to-end CBR system to generate the forecast text.

References

Adeyanju, I. 2012. Generating Weather Forecast Texts with Case based Reasoning. *International Journal of Computer Applications* 45(10):35–40.

Bellman, R. 1961. On the Approximation of Curves by line segments using dynamic programming. *Communications of the ACM* 4(6):284.

Belz, A. 2008. Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-Space Models. *Natural Language Engineering* 14(04):431–455.

Kolodner, J. L. 1992. An introduction to Case-Based Reasoning. *Artificial Intelligence Review* 6(1):3–34.

Sakoe, H., and Chiba, S. 1978. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26(1):43–49.

Sowdaboina, P. K. V.; Chakraborti, S.; and Sripada, S. 2014. Learning to summarize time series data. In *International Conference on Intelligent Text Processing & Computational Linguistics*, 515–528.

Sripada, S.; Reiter, E.; Hunter, J.; and Yu, J. 2002a. Segmenting time series for weather forecasting. *Applications and Innovations in Intelligent Systems X* 105–118.

Sripada, S.; Reiter, E.; Hunter, J.; and Yu, J. 2002b. Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data. *Computing Science Department, University of Aberdeen, Aberdeen, Scotland, Tech. Rep. AUCS/TR0201*.