A Framework for Generating Explanations from Temporal Personal Health Data

JONATHAN J. HARRIS, Rensselaer Polytechnic Institute
CHING-HUA CHEN, IBM Research, USA
MOHAMMED J. ZAKI*, Rensselaer Polytechnic Institute

Whereas it has become easier for individuals to track their personal health data (e.g., heart rate, step count, food log), there is still a wide chasm between the collection of data and the generation of meaningful explanations to help users better understand what their data means to them. With an increased comprehension of their data, users will be able to act upon the newfound information and work towards striving closer to their health goals. We aim to bridge the gap between data collection and explanation generation by mining the data for interesting behavioral findings that may provide hints about a user's tendencies. Our focus is on improving the explainability of temporal personal health data via a set of informative summary templates, or "protoforms." These protoforms span both evaluation-based summaries that help users evaluate their health goals and pattern-based summaries that explain their implicit behaviors. In addition to individual users, the protoforms we use are also designed for population-level summaries. We apply our approach to generate summaries (both univariate and multivariate) from real user data and show that our system can generate interesting and useful explanations.

CCS Concepts: • **Information systems** → **Data mining**; **Summarization**; *Data analytics*; *Personalization*; Recommender systems; • **Applied computing** → *Consumer health*; Health care information systems; Health informatics.

Additional Key Words and Phrases: linguistic data summarization, time-series analysis, sequence mining, natural language summaries, protoforms, personal health data

ACM Reference Format:
Jonathan J. Harris, Ching-Hua Chen, and Mohammed J. Zaki. 2020. A Framework for Generating Explanations from Temporal Personal Health Data. *ACM Trans. Comput. Healthcare* 1, 1 (March 2020), 30 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Smartphone apps and personal fitness devices have made it increasingly easy for users to collect and monitor their personal health data. While some of this data requires active entry by the user (e.g., dietary behaviors), other types of data are passively and continuously collected (e.g., physical activity, heart rate, location). The increased ease of data collection in the personal health domain has inspired the quantified-self movement, where motivated individuals record almost every aspect of their lives, including mental and physical health. Likewise, users with chronic conditions regularly use their own health information for health decision-making [35], and ineffective interpretation of one's data may adversely affect how they take their medications, what they eat, how they exercise and even how they socialize [30].

On the other hand, there are people who may not have a medical condition, or who are not quantified-selfers, but who simply wish to live a healthier lifestyle. For such people, it is widely reported that fitness devices and

Authors' addresses: Jonathan J. Harris, harrij15@rpi.edu, Rensselaer Polytechnic Institute, Troy, NY; Ching-Hua Chen, chinghua@us.ibm.com, IBM Research, Yorktown, NY, USA; Mohammed J. Zaki, zaki@cs.rpi.edu, Rensselaer Polytechnic Institute, Troy, NY.

2637-8051/2020/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

health apps experience a high abandonment rate. While there are many reported reasons for this, technology-related reasons include the lack of desired features such as *notifications* or *decision support*. Furthermore, if the user-perceived value of the data is low, this can create a feedback loop where such a perception increases the chance of erroneous or sparse data being recorded, which in turn lowers the utility of the data and leads to further user disengagement [10]. A key challenge for most users is often the lack of meaningful interpretation of the health data [9].

This concept can also be applied to personal health data. For those who may wish to improve or maintain their health, it is important for them to gain more insights into their own health logs to help them reach their personal health goals, and to assess whether their efforts are bringing them closer to those goals. However, individuals often find it challenging to understand their own health data, especially when they record multiple types of data over the long-term. For example, in the quantified-self community, structured recording of daily activities and outcomes is practiced regularly. A key hurdle for this community is the extraction of high-level information from the sea of data and to interpret the data in a meaningful way [9]. This hurdle is also commonly reported among patients living with chronic conditions [30], who use data for daily decision-making on medication dosages, food intake, and other behaviors. Ineffective interpretation of one's data may affect the subsequent decision-making process and anticipated health outcomes. The high frequency of data usage by those populations makes it impractical to rely solely on medical professionals to interpret their data. Automated methods to support data interpretation is therefore an urgent need.

Today, a common approach to obtaining expert-generated information on improving or maintaining health is through a search query online or through contact with a health expert. While searching for health information on the Internet works for the general case, it often lacks the personalization required to accommodate individual needs. In particular, every person has a different health experience, as exemplified by the uniqueness of the data collected by their health apps; human health experts may be able to relate an individual's data to general health knowledge, but they are expensive to engage and there are not enough of them. Therefore, health consumers are often left on their own to bridge the gap between the sea of general health knowledge and the sea of personal health data. Addressing this gap via automation requires a combination of methods for anticipating and understanding an individual's needs, providing an answer or recommendation for meeting that need, and, importantly, providing an explanation for that recommendation. While black box approaches that generate recommendations from data without explanation may be acceptable in some domains (e.g., manufacturing, advertising), this is rarely the case when it comes to personal health and healthcare. We believe that an important aspect of data-driven recommendation involves explaining how the data itself is being interpreted, and how it can be used to support explanations of downstream algorithms to produce a recommendation.

The main motivation of our work is the need of individuals (who wish to improve their health) to better understand their past behaviors based on their personal health data that may be inhibiting them from reaching their health goals. With additional comprehension via a natural language summary and a refined focus on key aspects of their lives, they will have the ability to take action by making appropriate changes to their routine. We address this problem by creating a framework that provides individuals with *personalized natural language summaries* based on behavioral patterns found within their time-series data. Generating explainable summaries from personal health data is a challenging task. Within the field of summarization, there are three main approaches when it comes to linguistic summary generation: probabilistic/statistical, neural, and rule-based methods [38]. Whereas state-of-the-art probabilistic/statistical and neural methods generate the sentences automatically, the textual output of these approaches is of lower quality than that of the rule-based approach [38]. Our work, therefore, utilizes a rule-based approach. Most existing methods within the linguistic summarization community do not handle time-series data; the few approaches that do either generate longer narratives of the data [14] or summaries of simpler trends [20], such as whether the trend is increasing, concave, etc. Our unsupervised approach takes advantage of a more

exhaustive set of summaries along with the use of time-series data mining methods to generate more meaningful explanations.

Our work focuses on improving the explainability of personal health data by generating temporal summaries in natural language from time-series health data. We propose a comprehensive framework to generate explanations that help a user evaluate their personal health data, and compare their data against general health guidelines or goals. In particular, we propose a systematic classification of summary types that cover a wide range of applications in personal health, including evaluation-based summaries that help users evaluate their health goals, and pattern-based summaries that explain their "hidden" behaviors. Our approach extracts temporal patterns from data and generates clear and concise explanations. In particular, our summaries are based on a categorical (or symbolic) representation of time-series data, combined with frequent sequence pattern mining and clustering, allowing us to generate understandable descriptions of hidden and implicit trends that are not obvious from the raw data. These trends may be found within and across multiple time series. This summarization framework mainly generates summaries by filling in sentence prototypes, or protoforms. An example protoform could be *On <quantifier> <sub-time window> in the past <time window>, your <attribute> was <summarizer>*, where the blanks (represented as *<blank>*) have certain constraints. For instance, the first blank requires what we call a quantifier. An example summary that can be generated from the protoform above is: *On most days in the past week, your step count was high.* Each summary explains how frequently a particular pattern is found for an attribute (or set of attributes) in the data, and what that could mean for a user.

With the generation of natural language summaries from both univariate and multivariate temporal personal health data, our system can provide important clues to a better understanding of a user's general behavior, and can facilitate actionable changes to fix areas where they may be falling short of their health goals. Finally, we showcase our framework on real user data from MyFitnessPal [39] food logs and the Insight4Wear [31] fitness dataset. To summarize, our work makes the following significant contributions:

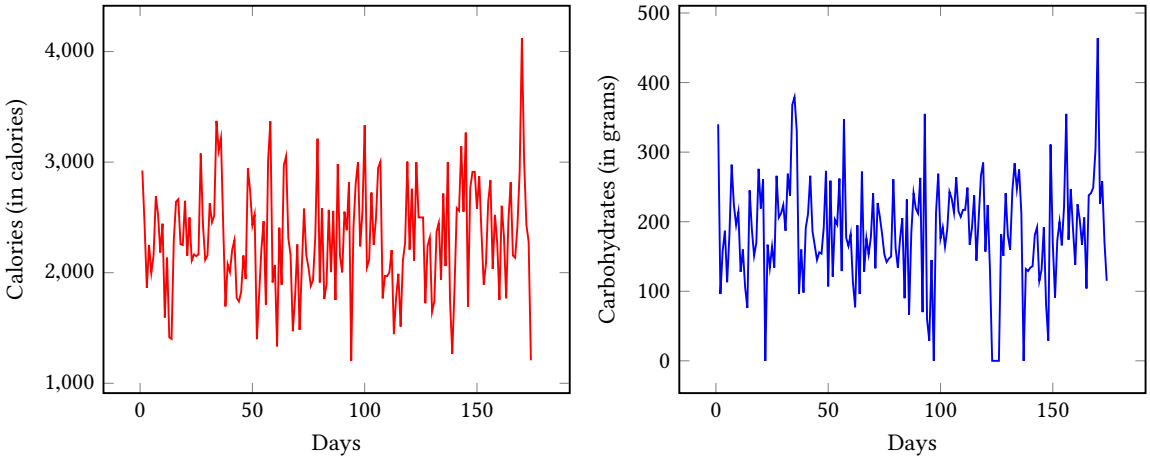
- We highlight interesting summaries obtained from users' nutrient intake, heart rate, and step count data.
- We introduce the utilization of pattern mining to drive protoform-based summarization approaches.
- We generate meaningful natural language summaries from both univariate and multivariate time-series data to highlight hidden patterns found within and between multiple variables.
- We propose a comprehensive framework of informative protoforms to produce both evaluation-based and pattern-based summaries using time-series data mining methods.
- We provide a systemic classification of summary types to be applied to the temporal personal health domain.
- We showcase and evaluate the usefulness of the summaries on data of food nutrient logs (using MyFitnessPal dataset [39]), and fitness data within the Insight4Wear [31] dataset.

2 TEMPORAL SUMMARIES FOR PERSONAL HEALTH DATA

We begin by defining basic concepts we will use in the remainder of this paper:

- **Protoform** (P): A sentence prototype (or template) that can be used to generate a natural language summary
- **Summarizer** (S): A conclusive phrase for the summary
- **Quantifier** (Q): A word or phrase that specifies how often the summarizer is true
- **Attribute** (A): A variable of interest
- **Time window** (TW): A time window of interest
- **Sub-time window** (sTW): A time window at a smaller granularity than TW
- **Qualifier** (R): a word or phrase that adds more specificity to a summary

Given a set of quantifiers **Q**, a set of summarizers **S**, a specified time window granularity TW and sub-time window granularity sTW, a set of protoforms **P**, and a set of time series **T** for a corresponding set of attributes **A**, we are able to generate natural language summaries of behavioral patterns found in temporal personal health data. In general, protoforms are of the form Q y's are S, where the quantifier $Q \in \mathbf{Q}$, $y \in \mathbf{A}$ and $S \in \mathbf{S}$ are special placeholders to be filled in by findings drawn from the data [44]. An example summary would be "*On most of the days, your calorie intake was high.*", where the quantifier Q ("most of the") represents how often the finding is found to be true in the data, the attribute y ("calorie intake") represents the variable of interest, and the summarizer S ("high") represents the conclusion from the data. Here the time window TW is "days." We use this general protoform as a basis to generate more complex summaries describing interesting patterns within and across variables.



(a) Calorie Intake Data

(b) Carbohydrate Intake Data

Fig. 1. Calorie (a) and Carbohydrates (b) Intake Data for a user from MyFitnessPal dataset [39]

Running Example: For example, consider a user from the MyFitnessPal dataset [39] who is keeping track of their intake of calories and carbohydrates for 174 days. The corresponding time series data is plotted in Figures 1a and 1b. Assume that the user is interested in finding patterns in a weekly time window and has the goal to limit their calorie and carbohydrate intake in a *2000-calorie* diet. For the remainder of this paper, we use this data as a running example to explain the various protoforms. We will also look at summary output using this example in Section 4. Denote time series **T** as the data corresponding to Figures 1a and 1b. In the next section, we will explore various protoform types where each protoform $p \in \mathbf{P}$ has a corresponding set of summarizers **S**. Table 2 enumerates the different types of summarizers (**S**), whereas Table 1 shows some of the possible quantifiers (**Q**). This table also shows the attributes of interest and the (sub-)time window values for our running example.

We seek to automatically generate a diverse set of explanations of time-series data related to a user's personal health. We propose different types of summaries, as shown in Figure 2, that are applicable in a wide range of personal health scenarios, and are meant to be both useful and comprehensive. We will first present protoforms for the summaries generated for a single user. These protoforms are also equally applicable to quantified-selfers or general users who want to understand their personal data. When a user looks at their own data, they may try to look for patterns in the data that correlate with their daily routine. These patterns reflect their behaviors and can provide clues as to what may be helping or hurting their progress towards their health goals.

Q	{ “none of the”, “almost none of the”, “some of the”, “half of the”, “more than half of the”, “most of the”, “all of the” }
A	{ “calorie intake”, “carbohydrate intake” }
TW	weekly granularity
sTW	daily granularity

Table 1. Variable Assignments

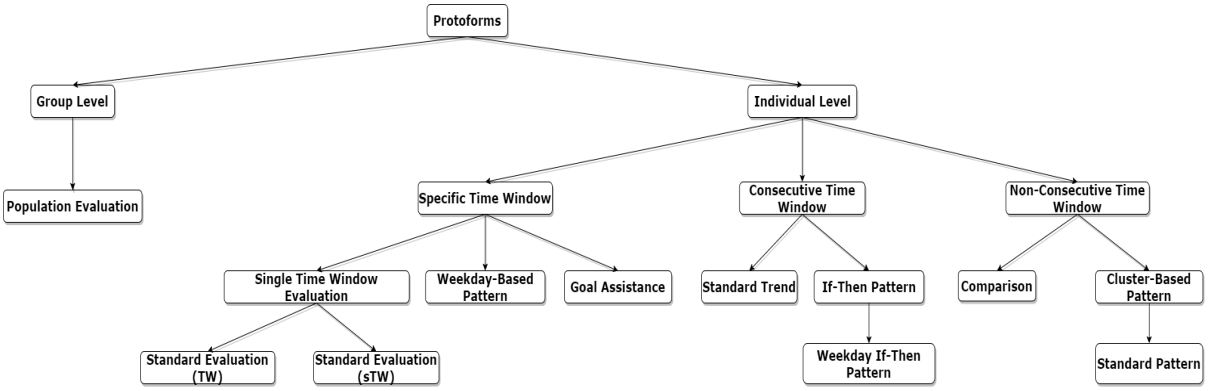


Fig. 2. Hierarchy of Summary Types

Table 2. Summarizers by Type

Protoform Type	Possible Summarizers
`Standard Evaluation`	very low, low, moderate, high, very high
`Standard Evaluation (w/ goal)`	reached, did not reach
`Goal Assistance`	increase, decrease
`Day-Based Pattern`	very low, low, moderate, high, very high
`Standard Trend`	increased, decreased, stayed the same
`If-Then Pattern`	very low, low, moderate, high, very high
`Comparison`	higher, lower, about the same
`Comparison (w/ goal)`	better, not do as well, about the same
`Cluster-Based Pattern`	rose, dropped, stayed the same

As illustrated in Fig. 2, we propose three types of individual-level summaries: 1) specific time window summaries, which look at a specified time window, 2) consecutive time window based summaries that compare two successive time periods, and 3) non-consecutive time window based summaries that compare different time periods. Each summary template requires a set of quantifiers and a unique set of summarizers as appropriate placeholders. In addition, these summaries can be augmented with goals or guidelines to better help the user. Below, we discuss each of our proposed summary types and provide univariate and multivariate examples using the data from the running example above. We will model the summary output using the given input variables.

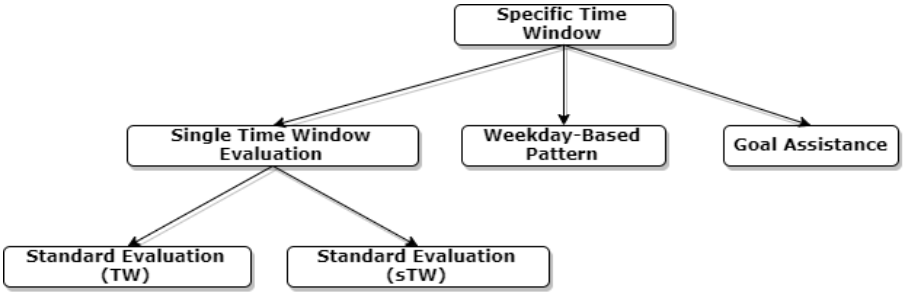


Fig. 3. Summary Types (Specific Time Window)

2.1 Specific Time Window Summaries

When looking for behavioral patterns in a time series, a user may want to search for patterns within a specific time window and to evaluate themselves within that time period. Since TW is set to a weekly granularity and sTW is set to a daily granularity in our running example, this user would be looking at a particular week (or days within that week) in their intake data so they can evaluate their progress. The summaries we generate for these patterns are called specific time window summaries; this sub-hierarchy of summaries is shown in Figure 3.

2.1.1 Standard evaluation summaries: Standard evaluation summaries are descriptions of evaluations made over the specified time window by pairing the standard evaluation summarizers from Table 2 with the "best" quantifier from Table 1. These summaries contain conclusions drawn from both TW and sTW and use summarizer set **S** = {*very low*, *low*, *moderate*, *high*, *very high*}.

Suppose the user is interested in knowing how well they have been doing for the past week. In order to find this information in **T**, the user may compare the past week with other weeks in the data. Our framework can generate standard evaluation summaries at the TW granularity to provide the user with this protoform:

> **Standard Evaluation Protoform** (TW granularity): In the past full <time window>, your <attribute 1> has been <summarizer 1>,..., and your <attribute *n*> has been <summarizer *n*>.
>
> **Univariate Example** (TW granularity): In the past full **week**, your **calorie intake** has been **high**.
> **Multivariate Example** (TW granularity): In the past full **week**, your **calorie intake** has been **high** and your **carbohydrate intake** has been **very high**.

Here *n* is the number of attributes. When the user receives these summaries on the TW (weekly) granularity, they are able to evaluate their past week as a whole relative to other weeks in their data. They can draw the conclusion that their intake of calories and carbohydrates has been relatively higher than previous weeks so they should work to minimize it for the next week. What happened during the past week? The user can switch to the sTW (daily) sub-time window granularity by looking at summaries modeled by this protoform:

> **Standard Evaluation Protoform** (sTW granularity): On <quantifier> <sub-time window> in the past <time window>, your <attribute 1> was <summarizer 1>,..., and your <attribute *n*> was <summarizer *n*>.
>
> **Univariate Example** (sTW granularity): On **some of the *days*** in the past **week**, your **calorie intake** has been **low**.
> **Multivariate Example** (sTW granularity): On **some of the *days*** in the past **week**, your **calorie intake** has been **low** and your **carbohydrate intake** has been **high**.

With these summaries, the user gains the knowledge that their calorie intake was actually low on some of the days in the past week. They can use these summaries to get a closer look and see if they can replicate their

behavior on those days. With this behavior change, they can get closer to their goals. The multivariate example implies that there may be a behavioral pattern between the user's calorie and carbohydrate intake. The user can explore this by using the following protoform enhanced with a qualifier:

> **Standard Evaluation Protoform** (w/ qualifier): On <quantifier> <sub-time window> in the past <time window> <qualifier>, your <attribute $n+1$> was <summarizer $n+1$>,...
>
> **Multivariate Example** (sTW granularity w/ qualifier): On **all of the *days*** in the past **week**, ***when your calorie intake was very low***, your **carbohydrate intake** was **moderate**.

For this summary, the user can clearly see a behavioral pattern that occurred in the past week. Whenever they had a very low calorie intake in the past week (the qualifier), their carbohydrate intake was moderate. They can use this summary to lower their carbohydrate intake if they choose to eat similar foods as on the day(s) they had a very low calorie intake. These summaries enable the user to comprehend how well they have performed in specific aspects (e.g., their calorie intake) within a specified time window.

2.1.2 Day-based pattern summaries: These summaries focus on patterns in the user's behavior in terms of certain attributes during "named" days of the week, e.g. Mondays. After receiving the standard evaluation summaries above, our user may wonder how they typically perform on certain days of the week. It is possible that they perform better for their health goals on certain days. The user can use the following protoform with **S** = {*very low*, *low*, *moderate*, *high*, *very high*}:

> **Day-Based Pattern Protoform**: Your <attribute 1> tends to be <summarizer 1>,..., and your <attribute n> tends to be <summarizer n> on <specified day>.
>
> **Univariate Example**: Your **calorie intake** tends to be **low** on **Thursdays**.
> **Multivariate Example**: Your **calorie intake** tends to be **very low** and your **carbohydrate intake** tends to be **very low** on **Thursdays**.

According to the summaries above, the user performs very well on Thursdays. For calorie intake alone, it is typically low on Thursdays; however, when looking at calorie and carbohydrate intake together, they are both typically very low on Thursdays. Using these conclusions, the user can monitor how they usually eat on Thursdays and try to emulate that on other days of the week.

2.1.3 Goal assistance summaries: Goal evaluation can be added to any summary type, to evaluate a certain attribute against a goal or a guideline. How would a user evaluate their progress towards their goals? If our user wishes to evaluate how well they limit their carbohydrate and calorie intake in a specific week, they can use this protoform:

> **Goal Evaluation Protoform**: On <quantifier> <sub-time window> in the past <time window>, you <summarizer 1> your goal to keep your <attribute 1> <goal 1>,..., and you <summarizer 1> your goal to keep your <attribute n> <goal n>.
>
> **Univariate Example**: On **most of the *days*** in the past **week**, you **did not reach** your goal to keep your **calorie intake *low***.
> **Multivariate Example**: On **some of the *days*** in the past **week**, you **did not reach** your goal to keep your **calorie intake *low*** and you **reached** your goal to keep your **carbohydrate intake *low***.

This protoform is similar to the one used for the standard evaluation summary at the sub-time window granularity, but with the summarizer set **S** = {*reached*, *did not reach*}. The user can use these summaries to realize that they fail to reach their calorie intake goal. On the bright side, they have some days where they reach their carbohydrate intake goals. These goals can be extracted from official health guidelines such as the ADA Lifestyle guidelines or suggested by health physicians [3].

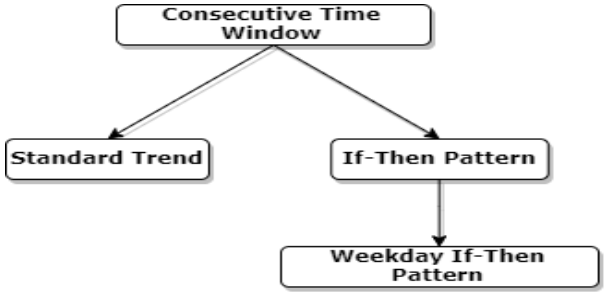


Fig. 4. Summary Types (Consecutive Time Window)

In addition to goal evaluation summaries, we also allow for goal assistance summaries that not only evaluate a user's progress towards a goal, but are also constructed to assist the user if they seem to be struggling. These summaries evaluate the user's data for multiple attributes against guidelines that are less defined, such as certain diets. The system must also determine which attributes to mention in the final summary without making the summary too lengthy. Goal assistance summaries can be thought of as a combination of goal evaluation summaries. This time, the set of summarizers is **S** = {*increase*, *decrease*}. If the user wishes to receive a more direct summary of what they should be working on, we can use this protoform:

> **Goal Assistance Protoform**: In order to better follow the <goal>, you should <summarizer 1> your <attribute 1>, <summarizer 2> your <attribute 2>, ..., and <summarizer n> your <attribute n>.
>
> **Univariate Example**: In order to better to follow the **2000-calorie diet**, you should **decrease** your **calorie intake**.
> **Multivariate Example**: In order to better to follow the **2000-calorie diet**, you should **decrease** your **calorie intake** and **increase** your **carbohydrate intake.**

Looking at the above summary, the user may actually want to increase their carbohydrate intake while lowering their calorie intake based on how they performed last week. This is an expected output as the *2000-calorie* diet recommends a higher amount of carbohydrates while the user wishes to limit their carbohydrate intake. It may be best for the user to switch instead to a *low-carbohydrate eating plan*.

2.1.4 General if-then pattern summaries: These summaries find possible correlations between multiple variables pertaining to a user's behavior over the entire time window. What if the user wishes to find a possible correlation between a certain behavior and an inhibiting action they take? The following protoform generates a summary that describes this correlation:

> **General If-Then Pattern Protoform**: In general, if your <attribute 1> is <summarizer 1>,..., and your <attribute n> is <summarizer n>, then your <attribute $n+1$> is <summarizer $n+1$>,..., and your <attribute $n+m$> is <summarizer $n+m$>
>
> **Example**: In general, if your **carbohydrate intake** is **very low** and your **fat intake** is **very low**, then your **calorie intake** is **very low**.

For our running example, there were no general if-then pattern summaries for only calorie and carbohydrate intake summaries using the data in Figures 1a and 1b. To provide an example, we factored in the user's fat intake as well in order to produce the above summary. These summaries have summarizer set **S** = {*very low*, *low*, *moderate*, *high*, *very high*}.

2.2 Consecutive Time Window Summaries

After searching within specific time windows to find behavioral patterns, our framework allows the user to move on to comparisons between time windows. Naturally, the user would start with consecutive time windows, or time windows that are next to each other. With TW set to a weekly granularity and sTW set to a daily granularity, our user will find patterns between consecutive weeks and consecutive days. The summaries below are referred to as consecutive time window summaries, and their relationships are shown in Figure 4.

2.2.1 Standard trend summaries: Suppose the user wishes to know how they perform from day to day. Looking at the data, how often does their calorie intake increase or decrease between days? We can use a standard trend summary to see this.

Standard trend summaries describe trends from one sub-time window to the next. These summaries can be used to describe a user's tendency between two consecutive sub-time windows and use summarizer set **S** = {*increased*, *decreased*, *stayed the same*}.

> **Standard Trend Protoform**: <Quantifier> time, your <attribute 1> <summarizer 1>,..., and your <attribute *n*> <summarizer *n*> from one <sub-time window> to the next.
>
> **Univariate Example**: **Half of the** time, your **calorie intake *increases*** from one **day** to the next.
> **Multivariate Example**: **Some of the** time, your **calorie intake *increases*** and your **carbohydrate intake *increases*** from one **day** to the next.

These two summaries allows our user to know that there is around a 50% chance that their calorie intake will increase the next day and that there is a relatively smaller chance that their calorie intake and their carbohydrate intake will both increase the next day. While similar to standard evaluation summaries (which evaluate the attribute on each day), here we evaluate the attribute between one day and the next; these summaries are ratio-based and span the entire dataset, instead of a specified time window.

2.2.2 If-then pattern summaries: What if the user wishes to know more about how their past and current behaviors predict the trends in the near future? For answering this, we propose if-then pattern summaries, that provide more interesting patterns based on frequent sequence mining [45]. These patterns span multiple consecutive sub-time windows and are of variable length, constrained by the size of the time window. They use summarizer set **S** = {*very low*, *low*, *moderate*, *high*, *very high*}. The protoform is:

> **If-Then Pattern Protoform**: There is <confidence value> confidence that, when your <attribute 1> is <summarizer 1:1>, then <summarizer 2:1>,..., then <summarizer *m*:1>,..., and your <attribute *n*> is <summarizer 1:*n*>, then <summarizer 2:*n*>,..., then <summarizer *m*:*n*>, your <attribute 1> tends to be <summarizer ($m+1$):1>,..., and your <attribute *n*> tends to be <summarizer ($m+1$):*n*> the next <time window>.
>
> **Univariate Example**: There is **100%** confidence that, when your **calorie intake** follows the pattern of being **moderate**, your **calorie intake** tends to be **very low** the next **day**.
> **Multivariate Example**: There is **100%** confidence that, when your **calorie intake** follows the pattern of being **very low**, and your **carbohydrate intake** follows the pattern of being **moderate**, your **carbohydrate intake** tends to be **moderate** the next **day**.

In the above protoform, m represents the number of summarizers per attribute and n represents the total number of attributes. From these summaries, the user can conclude that their calorie intake is typically very low after having a moderate intake the previous day. On top of that, if they have a very low calorie intake and a moderate carbohydrate intake on the same day, they will typically maintain the same moderate carbohydrate intake the next day.

How about if the user wants to see these behavioral patterns pertaining to days of the week? If-then pattern summaries can also be made dependent on the day of the week, via the protoform:

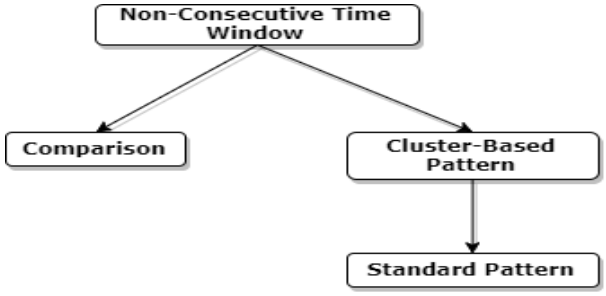


Fig. 5. Summary Types (Non-Consecutive Time Window)

> **Day If-Then Pattern Protoform**: There is <confidence value> confidence that, when your <attribute 1> is <summarizer 1:1> on a <day 1:1>, then <summarizer 2:1> on a <day 2:1>,..., then <summarizer *m*:1> on a <day *m*:1>,..., and your <attribute *n*> is <summarizer 1:*n*> on a <day 1:*n*>, then <summarizer 2:*n*> on a <day 2:*n*>,..., then <summarizer *m*:*n*> on a <day *m*:*n*>, your <attribute 1> tends to be <summarizer (*m* + 1):1> the next <day (*m* + 1):1>,..., and your <attribute *n*> tends to be <summarizer (*m* + 1):*n*> the next <day (*m* + 1):*n*>.
>
> **Univariate Example**: There is **100%** confidence that, when your **calorie intake** follows the pattern of being **very high** on a **Sunday**, your **calorie intake** tends to be **low** the next **Monday**.
> **Multivariate Example**: There is **100%** confidence that, when your **calorie intake** follows the pattern of being **very high** on a **Saturday**, then **very high** on a **Sunday**, your **carbohydrate intake** tends to be **very high** the next **Sunday**.

In the above protoform, *m* represents the number of summarizer-day pairs per attribute and *n* represents the total number of attributes. Looking at these patterns for similar days, the user can see that they have a low calorie intake on Mondays following a very high intake the previous Sunday. Also, if they have weekends with very high calorie intake, they typically consume a lot of carbohydrates on the next Sunday.

2.3 Non-Consecutive Time Window Summaries

Having examined the patterns that can be found between consecutive time windows, the user may try to find patterns across time windows that are not consecutive. Perhaps the past week they had was similar to another week that occurred a month earlier in the data. Summaries explaining these types of patterns are called non-consecutive time window summaries. These summaries look at time windows that do not necessarily have to be consecutive; they compare discovered trends found in one time window with those of another time window in the data.

2.3.1 Comparison summaries: What if the user wants to make comparisons between their most recent week of logging and a week in a much earlier part of their data? Comparison summaries provide comparisons between any two different time windows to help users evaluate their behavioral differences. These summaries use summarizer set **S** = {*higher*, *lower*, *about the same*}. The protoform is:

> **Comparison Protoform**: Your <attribute 1> was <summarizer 1>,..., and your <attribute *n*> was <summarizer *n*> on <time window 1> <number 1> than they were on <time window 2> <number 2>.
>
> **Univariate Example**: Your **calorie intake** was **higher** in **week *23*** than it was in **week *11***.
> **Multivariate Example**: Your **calorie intake** was **higher** and your **carbohydrate intake** was **higher** in **week *23*** than they were in **week *11***.

Looking at the past week and another week earlier in the data, the user can see that their intakes of calories and carbohydrates of this past week were higher than they were three months before. These summaries show the user that they may need to try and replicate what they did the previous week to get closer to their health goals.

Comparison summaries can also be enhanced with a goal using summarizer set **S** = {*better*, *not do as well*, *about the same*}. We display the protoform below:

> **Goal Comparison Protoform**: You did <summarizer 1> overall with keeping your <attribute 1> <goal 1> ,..., and you did <summarizer *n*> overall with keeping your <attribute *n*> <goal *n*> in <time window 1> <number 1> than you did in <time window 2> <number 2>.
> **Univariate Example**: You did **better** overall with keeping your **calorie intake *low*** in **week *23*** than you did in **week *11***.
> **Multivariate Example**: You did **better** overall with keeping your **calorie intake *low*** and you did **better** overall with keeping your **carbohydrate intake *low*** in **week *23*** than you did in **week *11***.

2.3.2 Cluster-based pattern summaries: The user may also wish to predict how they will act the following week based on their behavior in the past week. One method to achieve this would be to find other weeks most similar to this past one and to see what happened in the following weeks. We use cluster-based pattern summaries to display these patterns.

These summaries factor in all of the other time windows that are similar to the time window in question, resulting in a cluster. For example, if we are looking at the current week, our system will factor in every other week that has a similar representation (using the Squeezer [17] clustering algorithm). These summaries use summarizer set **S** = {*rose*, *dropped*, *stayed the same*}. In addition to a protoform, we also add a description of the preceding week.

> **Preceding Time Window Description Protoform**: In <time window> <week number>, your <attribute 1> was <summarizer 1:1>, then <summarizer 2:1>,..., then <summarizer m_1:1>,..., and your <attribute *n*> was <summarizer *n*:1>, then <summarizer *n*:2>,..., then <summarizer m_n:*n*>.
> **Cluster-Based Pattern Protoform**: During <quantifier> <time window (plural)> similar to <time window> <week number>, your <attribute 1> <summarizer 1>,..., and your <attribute *n*> <summarizer *n*> the next <time window>.
>
> **Univariate Example**: In **week *11***, your **calorie intake** was **moderate**, then **very low**, then **high**, then **very high**, then **low**, then **moderate**. During **all of the *weeks*** similar to **week *11***, your **calorie intake *rose*** the next **week**.
> **Multivariate Example**: In **week *11***, your **calorie intake** was **moderate**, then **very low**, then **high**, then **very high**, then **low**, then **moderate** and your **carbohydrate intake** was **moderate**, then **high**, then **very low**, then **high**. During **all of the *weeks*** similar to **week *11***, your **calorie intake *rose*** and your **carbohydrate intake *rose*** the next **week**.A

Here m_i is the number of summarizers for attribute i, and n is the number of attributes. Note that the quantifier is calculated from the cluster alone instead of the entire dataset. We can see that, in every summary of this type, the description of the time window comes first. The description is then followed by the actual protoform. From these summaries, the user in our running example is able to know how exactly their past week went for each nutrient. The user can also conclude that their calorie intake will likely rise the next week, while their carbohydrate intake has around a 50% chance of dropping. The user can focus on limiting their calorie intake the next week to avoid repeating this pattern.

The user may also wish to focus on the most recent week. Despite the conclusions stated by the cluster-based pattern summaries, it is possible that the user has not behaved this way recently. Cluster-based pattern summaries can also be used for what we call a standard pattern protoform:

> **Standard Pattern Protoform**: The last time you had a <time window> similar to <time window> <number>, your <attribute 1> <summarizer 1>,..., and your <attribute *n*> <summarizer *n*> the next <time window>.
> **Univariate Example**: The last time you had a **week** similar to **week *11***, your **calorie intake *rose*** the next **week**.

> **Multivariate Example**: The last time you had a **week** similar to **week *11***, your **calorie intake *rose*** and your **carbohydrate intake *rose*** the next **week**.

The user can now see from the standard pattern summaries that (for the most part) the behavior found by the cluster-based pattern summaries still repeats in the most recent week. This reinforces the user's motivation to work towards changing this behavior during the next week.

2.4 Group-Level Summaries

After looking at their own data, what if the user was curious about how the entire user population is faring as a whole or how the user compares to other users (where such data is available)? We can use group-level summaries to find this answer.

Population evaluation summaries. For now, the group-level summary our system can generate is the population evaluation summary (see Figure 2). This can be used to summarize the study population as a whole using the individual summaries previously generated by our system. If the user wishes to know how they compare against other users in terms of their calorie and carbohydrate intake in the past week, they can use this protoform:

> **Population Evaluation Protoform**: <Quantifier 1> users in this study had a <summarizer 1> <attribute 1>, a <summarizer 2> <attribute 2>,..., and a <summarizer *n*> <attribute *n*> <sub-protoform>.
> **Univariate Example (Standard Evaluation (TW))**: **Some of the** users in this study had a **moderate *calorie intake* in the past full week**.
> **Multivariate Example (Standard Evaluation (TW))**: **Some of the** users in this study had a **high *calorie intake*** and a **very high *carbohydrate intake* in the past full week**.

Here we define <sub-protoform> as a portion of an actual summary used to describe a number of users in the dataset. This "sub-protoform" identifies the summary type the population has been evaluated on and the conclusion found. The user in our running example can use these summaries to know that at least some of the users in the study did better at managing their calorie intake in the past full week. Now, the user may be more motivated to make changes in their diet for the upcoming week.

3 SUMMARY GENERATION AND MINING

In this section, we provide a description of our summary generation approach.

Representing Time Series as Symbolic Sequences. In order to draw insights from the time-series data, such as frequent patterns and anomalistic behavior, we first represent the raw time-series data in symbolic form. To achieve this, we use the SAX symbolic representation [25] for each time series, which also makes it easier to represent the time series at different granularities. Figure 6 below provides a visualization of how each time series is turned into a string of symbols.

The symbols, in particular, are letters from some alphabet. Provided an alphabet size n and the time window size, SAX z-normalizes the raw data of each time series to a zero mean with a standard deviation of 1. It then uses Piecewise Approximate Aggregation (PAA) to reduce the dimensionality of each time series, depending on the time window size. This reduction allows the ability to easily switch between granularities. After the data is projected onto its principle components and normalized, SAX generates n equiprobability bins based on the standard Gaussian distribution with each segment represented by its corresponding bin symbol. Fig. 6 shows an example time-series curve (blue line), with an alphabet size of $n = 3$. Letters are assigned to the bins in alphabetical order from the lowest bin to the highest bin (see right hand y-axis), with the bins generated from the Gaussian (see left hand y-axis). This time series is represented as the symbolic sequence "baabccbc", with each element corresponding to the letter for the window segment bin.

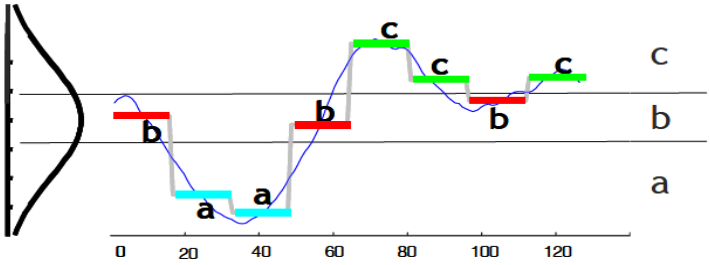


Fig. 6. SAX Representation (from [25])

3.1 Pattern Mining

We employ two different types of pattern mining approaches, based on clustering and frequent sequence mining.

Cluster-based patterns. After partitioning the sub-time window SAX representation into time window tuples (e.g., chunking a string of days into weeks), we combine multiple time series into multivariate symbolic sequences. For example, if one variable has the SAX sequence "abacbbc" and another the sequence "bccabcc", then the combined multivariate sequence is "a-b, b-c, a-c, c-a, b-b, b-c, c-c", where the symbols in the corresponding positions have been combined into an "event." We then group these combined sequences into clusters. For clustering, we use Squeezer [17], which is an online clustering algorithm for categorical data that only needs a similarity threshold s to find clusters. For each tuple t, Squeezer assigns it to an existing cluster or creates a new cluster based on the similarities between t and the existing clusters, using threshold s. We use a sampling-based approach to determine s. We sample a fraction f of the window tuples and calculate the average similarity between each pair of tuples in the sample, using

$$sim(T_i; T_j) = \left|\{A_k | T_i.A_k = T_j.A_k, 1 \leq k \leq n\}\right| \tag{1}$$

where T_i and T_j are tuples, A_k is the SAX symbol at index k, and n is the time-window size, i.e., the similarity is based on the number of matching symbols at corresponding positions. We repeat this process $1/f$ times (e.g., if we sample $f = 0.2$ or 20% of the tuples, we repeat the sampling $1/f = 5$ times), and set a as the mean of all the average pair-wise similarities. Finally, we set $s = a + 1$, as suggested in [17].

Each cluster now contains non-consecutive time windows that have been grouped together by similarity. From these clusters we can use the history of the attributes involved to "predict" what may happen in the time window following the one we are interested in. Typically we choose the most recent time window in an attempt to "predict" the future, although it may also be beneficial to use another time window. If we were to use a time window other than the most recent one, we can extract the expected result for the following time window. In short, if we have a time window TW, we should be able to use the result of similar time windows to see the expected outcome of the time window following TW.

For each cluster, we pair each tuple with the tuple that follows it, e.g., pair each week in a cluster with the week following it. Next, we replace the tuples, which are at the sub-time window level (sTW), with the time-window level (TW) SAX symbols. These time-window level pairs are used to generate cluster-based pattern summaries. In order to describe a pattern, we map the letters (typically, in the last full week) to their corresponding summarizers.

Frequent sequence mining for if-then patterns. To generate if-then pattern summaries, we employ frequent sequence mining over the symbolic SAX temporal data using SPADE [45]. "Frequent" means that the pattern

appears more than a user-specified value called "minimum support." The method outputs all of the frequent sequence patterns found in the data.

For each frequent sequence, we map each of its prefixes to the following suffixes. For instance, if "abca" is a frequent sequence, then we consider the pairs: ('a','bca'), ('ab','ca'), and ('abc','a'). A similar approach is taken for multivariate data. Next, we generate confidence values (or conditional probability) of observing the suffix given the prefix, given as

$$P(\text{suffix}|\text{prefix}) = \frac{\text{count(prefix + suffix)}}{\text{count(prefix)}} \tag{2}$$

where count(seq) is the frequency of the sequence "seq." We use a minimum confidence threshold to retain only those frequent if-then patterns (of the form "If prefix then suffix") with the highest confidence values. Finally, these patterns are used to generate summarizers to be presented to the user.

3.2 Explanation Generation

In order to generate summaries, we fill in the blanks of protoforms presented in Section 2 using summarizers from Table 2 and quantifiers from Table 1. The dataset is also modified depending on the summary type. For instance, for standard evaluation, the dataset is the past full week of the data.

As there are many possible combinations of summarizers and quantifiers for each attribute, we choose a combination that is "most appropriate," based on the *average membership function* for a summarizer S and a quantifier Q. We denote μ_S as the membership function value for summarizer S in a time window TW. The membership value μ_S will either have the value of 1 or 0, based on whether the value v of the time window for attribute A follows the conclusion implied by the summarizer. For example, when evaluating a goal for calorie intake where a user wishes to eat at most 2,000 calories a day, the possible summarizers would be "reached" or "did not reach", according to Table 2 (for the standard evaluation summary with a goal). In this case, a value v less than or equal to 2,000 would imply that the user "reached" the goal, while a value v greater than 2,000 would imply that the user "did not reach" the goal. From the μ_S for each time window, we calculate the aggregated average

$$r_S = \frac{1}{n}\sum_{i=1}^{n} \mu_S(y_i) \tag{3}$$

where y_i is a data point in the dataset and n is the size of the dataset. This fraction r_S indicates the percentage of the dataset that agrees with the summarizer S.

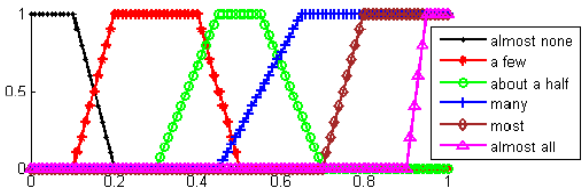


Fig. 7. Adapted from [19]

Once we obtain the r_S value for each summarizer S, we use this value to determine the best quantifier for each summarizer. We employ the use of trapezoidal membership functions [42] μ_Q to calculate how well r_S fits each quantifier. As μ_S is the membership function of a summarizer S, μ_Q is the membership function of a quantifier Q.

For example, in Figure 7, for the quantifier "most" ("most of the" in our framework), we have

$$\mu_Q(r_S) = \begin{cases} 4r_S - 2 & 0.5 < r_S < 0.75 \\ 1 & 0.75 \leq r_S \leq 0.9 \\ -10r_S + 10 & 0.9 < r_S < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We have manually defined membership functions for each possible quantifier based on the approach in [21]. They were designed to create trapezoidal functions that match with the values we believe best fit each quantifier.

Once we have the best quantifier for each summarizer, we will have k quantifier-summarizer candidate pairs. The pair that contains the quantifier with the highest μ_Q will be chosen for the summary, while the value of μ_Q eventually becomes the summary's *truth value*. When the most appropriate summarizer and quantifier are found, they are used within the protoform or template to generate the explanation. As a result, we generate a list of candidate summaries, paired with a truth value using the μ_Q value of the quantifier within the summary [43]. Finally, we choose the summary with the highest truth value μ_Q, breaking ties by selecting the summary with the quantifier that implies the largest amount (following Yager's approach for text quality [42]).

3.2.1 Summary Metrics. To evaluate the summaries generated by our system, we use five evaluation metrics, which help measure some of the Gricean maxims [15]: the maxims of quality, quantity and relevance. These maxims are well-known pragmatic rules that are known to improve communication of information to humans [6], especially for natural language summaries. The evaluation metrics we use on our summaries are the degrees of truth, imprecision, covering, appropriateness, and coverage.

Degree of Truth: First and foremost, we want the summaries our framework generates to convey the degree of truth. We use natural language to summarize how often a finding may be true in the data. We use fuzzy quantifiers to describe the frequencies of certain behaviors that best fit the percentage found in the data, although the truthfulness of the overall summary may not be absolute.

Zadeh's degree of truth [43] determines which summaries are actually true statements. We use this degree to measure to what extent our summaries follow the maxim of quality, which states how true a summary is and how much evidence supports it. We have already discussed how to calculate this in Section 3.2, where we use (Eq. 3) to calculate the ratio of the dataset that supports the summarizer S. Then, we calculate the μ_Q for the quantifier in each quantifier-summarizer pair via (Eq. 4), which becomes the summary's truth value. As we use μ_Q to select the best quantifier-summarizer pair, we can simply reuse the value as the truth value. For the remainder of this paper, we will refer to the degree of truth as T_1.

Degree of Imprecision: Also known as the degree of fuzziness, the degree of imprecision measures how useful a summary is. It is highly possible that a summary is generated that has a high degree of truth but is also a statement that is not useful, such as "All winter days are cold."

Recall that r_S indicates the fraction of the dataset that agrees with the summarizer S. In our summary generation approach, we keep track of percentages r_{S_j} of each possible summarizer S_j. In order to calculate the degree of imprecision, we use the following equation:

$$I = 1 - \sqrt[m]{\prod_{j=1\ldots m} r_{S_j}} \tag{5}$$

where m is the number of possible summarizers for the protoform type. Here we compute the geometric mean of the percentages of agreement over the possible summarizers S_j. For the case where a summary is obvious, every summarizer S_j would have a membership value $\mu_{S_j} > 0$ for every (sub-)time window. For example, if every day was snowy and cold, then it would be unwise to output a summary describing this trend. When we subtract the

geometric mean from 1, the degree of imprecision represents the extent to which the summary is useful. For the remainder of this paper, we will refer to the degree of imprecision as T_2.

Degree of Covering: The degree of covering C measures how many objects in a user's query are covered by the summary. When we refer to the user's query, we are referring to the subset d of the dataset D that is used to create the summary. We wish to know how often the summary's conclusion (the summarizer S) is true within the subset d. Within this domain, the degree of covering can simply be expressed as the r_S of the summarizer S of the summary restricted to d. When we generate summaries, we specify time windows to look for particular trends depending on the protoform type. In this way, the time window specification is the "query" whereas the time window itself is the subset d of D. We already base the ratio r_S off of how often summarizer S is true in d so we are already calculating C. In other words,

$$C = r_S \tag{6}$$

where S is the best summarizer for the summary. Although this degree is just the ratio of the subset d that agrees with the summary, it is useful to see the actual percentage of agreement alongside the summary. In cases where the quantifier's definition is more fuzzy (i.e., the range of the trapezoidal membership function is especially large), it may be useful to know the exact percentage. For example, if a summary uses the "some of the" quantifier, the trapezoidal function corresponding to this quantifier ranges from an agreement of 10% to 50% of d. Even if the quantifier is not guaranteed to be chosen unless the percentage is between 30% and 40%, it is still useful to know what the actual percentage is. For the remainder of this paper, we will refer to the degree of covering as T_3.

Degree of Appropriateness: The degree of appropriateness [21] also helps avoid trivial summaries. The degree's value represents how interesting and unexpected a finding in the summary may be. We use this degree to measure to what extent our summaries follow the maxim of quantity, which states how much information should be conveyed in a summary. When communicating with a human user, it is important that our summaries avoid providing: 1) too much information to easily process when reading the summary, or 2) too little information to fully comprehend the findings implied and to act upon those findings.

To calculate the degree of appropriateness of a summary, the summary is split into K sub-summaries by attribute. For each sub-summary, the percentage r_k of the data where the membership value is $\mu_{S_k} > 0$ is calculated, with r_k given as:

$$r_k = \frac{1}{n} \sum_{i=1}^{n} \mu_{S_k}(y_i) \tag{7}$$

Afterwards, the product

$$r^* = \prod_{k=1}^{K} r_k \tag{8}$$

of the percentages r_k is calculated. Finally, the absolute difference between r^* and the summary's degree of covering C, given as

$$A = |r^* - C| \tag{9}$$

yields the degree of appropriateness.

This degree is mainly used to analyze relations in the data. For example, if a user has a high calorie intake on 50% of the days and a low carbohydrate intake on 50% of the days, one may expect that the user has a high calorie intake and a low carbohydrate intake on 25% of the days. This intuition corresponds to the product of ratios r^* above. If, however, the actual percentage of days differs from 25%, then we can say that the outcome is unexpected and the difference represents the extent to which the outcome differs from what was expected. In terms of the level of informativeness, if the degree of appropriateness is 0, it is possible that the summary states too much where the finding is very precise or too little where the finding is very vague. For the remainder of this paper, we will refer to the degree of appropriateness as T_4.

Degree of Coverage: Finally, we want the summaries to be relevant to the user. It would not be very useful to receive a summary that is not very relevant to a user's context or situation. The maxim of relevance states how relevant the summary should be. It is calculated by using the degree of coverage C^* (not to be confused with C, the degree of covering), which determines whether the conclusion made by the summary is supported by enough data [41]. If the summary is not supported by enough data, then it may not be worth stating.

We can use the ratio r_S to find the percentage of the data that agrees with the summary. The degree of coverage [41] is given as:

$$C^* = f(r_S) = \begin{cases} 0 & r_S \leq r_1 \\ \dfrac{2(r_S - r_1)}{(r_2 - r_1)^2} & r_1 < r_S < \dfrac{r_1 + r_2}{2} \\ 1 - \dfrac{2(r_S - r_1)}{(r_2 - r_1)^2} & \dfrac{r_1 + r_2}{2} \leq r_S < r_2 \\ 1 & r_S \geq r_2 \end{cases} \tag{10}$$

In the equation above, Wu et al. [41] use values of 0.02 and 0.15 for r_1 and r_2, respectively. The definition of this function creates an curve similar to the trapezoidal membership function of μ_Q for the quantifier "almost all" in Figure 7. Where r_S lies on this curve determines how relevant the finding is. For the remainder of this paper, we will refer to the degree of coverage as T_5.

4 EXPERIMENTS

We ran experiments on multiple datasets to analyze the different types of protoforms we generate. In particular, we use real data from the *MyFitnessPal Food dataset* [39], which consists of 587,187 days of food data across 9,900 users over a course of up to 180 days. Each entry logs a user's food items with nutrient information, daily totals, and user's nutrient goals. We also use user health data from *Insight4Wear* [31], which is a quantified-self/life-logging app, with about 11.5 million records of information. It provides data gathered from mobile devices that track step count, heart rate, and user activities for around 1,000 users.

For all of the example output reported earlier in the paper, we used a default alphabet size of $n = 5$, a time window of seven days, a minimum support of 20%, and a minimum confidence of 80%, which comprise the default parameter values. We explore the use of different sets of input parameters later in this section. Below, we provide results of our framework's summary generation on real user data, and also show quantitative results in terms of evaluation metrics. It is important to note that existing systems for summary generation are either not publicly available, or they do not handle time-series data; therefore, a direct comparison is not feasible. Nevertheless, we qualitatively showcase how our framework compares to other state-of-the-art works on temporal data from stock market and weather domains. On the other hand, for reproducibility, our implementation is open source, and can be downloaded from https://github.com/harrij15/TemporalSummaries.

4.1 Summary Generation

We show summaries on calorie and carbohydrate intake from the MyFitnessPal food log dataset and heart rate data from Insight4Wear. All summaries are generated using the default input parameters.

4.1.1 Calorie and Carbohydrate Intake: MyFitnessPal Food Logs. Based on the calorie and carbohydrate data shown in Figures 1a and 1b, we display three lists of summaries generated for our user: one for calorie intake (univariate), one for carbohydrate intake (univariate), and another for summaries handling both calorie and carbohydrate intake (multivariate). For each list, there are also corresponding group-level summaries evaluated on 389 users (15,915 summaries) from the food log dataset. We selected users that have logged at least 175 days. In total, our system generates 113 summaries.

Calorie Intake: Univariate Summaries. With the calorie intake data, we are able to use the summaries to draw a picture of how the user usually handles their calories and what kind of conclusions can we draw from this data and how does our user compare to the rest of the study population. Our system produces 19 individual-level summaries using 11 protoforms and 16 group-level summaries using 5 protoforms. To avoid repetition, 11 representative individual-level summaries are shown in Table 3 and 9 group-level summaries are shown in Table 4.

Table 3. Univariate Individual-Level Summaries for Calorie Intake Data

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
`Standard Evaluation (TW)`	In the past full week, your calorie intake has been high.	N/A	N/A	1	0.75	1
`Standard Evaluation (sTW)`	On some of the days in the past week, your calorie intake has been low.	0.93	0.81	0.29	0.05	1
`Standard Evaluation + Goal`	On most of the days in the past week, you did not reach your goal to keep your calorie intake low.	1	0.65	0.86	0.11	1
`Comparison`	Your calorie intake was higher than it was the week before.	N/A	N/A	1	0	1
`Comparison + Goal`	You did not do as well overall with keeping your calorie intake low than you did the week before.	N/A	N/A	1	0	1
`Standard Trend`	Half of the time, your calorie intake increases from one day to the next.	0.71	0.84	0.53	0	1
`Cluster-Based Pattern`	This past week, your calorie intake was moderate, then very low, then high, then very high, then low, then moderate. During more than half of the weeks similar to this past one, your calorie intake rose the next week.	1	0.71	0.6	0.3	1
`Standard Pattern`	The last time you had a week like this past one, your calorie intake rose the next week.	N/A	N/A	1	0.7	1
`If-Then Pattern`	There is 100% confidence that, when your calorie intake follows the pattern of being moderate, your calorie intake tends to be very low the next day.	N/A	N/A	0.53	0.26	0.32
`Day If-Then Pattern`	There is 100% confidence that, when your calorie intake follows the pattern of being very high on a Sunday, your calorie intake tends to be low the next Monday.	N/A	N/A	0.53	0.8	0.2
`Day-Based Pattern`	Your calorie intake tends to be low on Mondays.	1	0.82	0.36	0.12	1
`Goal Assistance`	In order to better to follow the 2000-calorie diet, you should decrease your calorie intake.	N/A	N/A	N/A	N/A	N/A

Table 4. Univariate Group-Level Summaries for Calorie Intake Data

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
`Standard Evaluation (TW)`	Some of the users in this study had a low calorie intake in the past full week.	1	0.84	0.35	0	1
`Standard Evaluation (sTW)`	Almost none of the users in this study had a low calorie intake on more than half of the days in the past week.	1	0.98	0.1	0	0.21
`Standard Evaluation + Goal`	Some of the users in this study reached their goal to keep their calorie intake low on all of the days in the past week.	0.84	0.87	0.42	0	1
`Comparison`	Some of the users in this study had a similar calorie intake than they did the week before.	1	0.68	0.36	0	1
`Comparison + Goal`	Some of the users in this study did about the same with keeping their calorie intake low than they did the week before.	1	0.68	0.36	0	1
`Standard Trend`	More than half of the users in this study increase their calorie intake from one day to the next half of the time.	0.91	1	0.59	0	1
`Cluster-Based Pattern`	After looking at clusters containing weeks similar to this past one, it can be seen that some of the users with these clusters may see a rise in their calorie intake next week.	1	0.67	0.37	0	1
`Standard Pattern`	Based on the most recent weeks similar to this past one, it can be seen that some of the users may see a drop in their calorie intake next week.	1	0.67	0.32	0	1
`Day-Based Pattern`	Some of the users in this study tend to have a low calorie intake on Mondays.	0.81	1	0.26	0	1
`Goal Assistance`	All of the users in this study have been given advice to decrease their calorie intake.	1	0	1	0	1

From the individual-level summaries, it becomes apparent that the user is struggling with their calorie intake. The standard evaluation (TW) and goal evaluation summaries explain how our user has struggled in the past

week. We can gather from the comparison and goal comparison summaries that the user is also performing worse than the week before, so they are getting further from their health goals. From the standard trend, the cluster-based pattern, and the standard pattern summaries, our user can also see that they will most likely do even worse the next week unless they make changes to their usual routine. In order to make changes, our user can look at the standard evaluation (sTW), the (day) if-then pattern, and the day-based pattern summaries to closely look at their behavioral tendencies and see when they did things right. Looking at the group-level summaries, the user seems to be performing as well as some of the other users when comparing the summaries, although performing fairly worse than the average user.

When looking at the evaluation metrics in these tables, we can see that some of the evaluation metrics do not apply to all of the individual-level summary types (labeled as N/A). For the degrees of truth T_1 and imprecision T_3, there are certain individual-level summary types that do not use a ratio-based method on a subset d of the dataset D. i The goal assistance summary is the only type that does not apply to any of the summary metrics as it is depends only on the average value of the last week's values in order to draw conclusions about how well the user followed a certain diet in the past full week.

Carbohydrate Intake: Univariate Summaries. For carbohydrate intake data, our system produces 19 individual-level summaries using 9 protoforms and 14 group-level summaries using 6 protoforms. The conclusions we can draw from the carbohydrate intake are very similar to what we drew from the calorie intake (using the same protoforms). At the group level, the user can know that most of the other users struggled to reach their daily carbohydrate intake goals in the past week. We omit the detailed results since they are qualitatively similar to the calorie intake case.

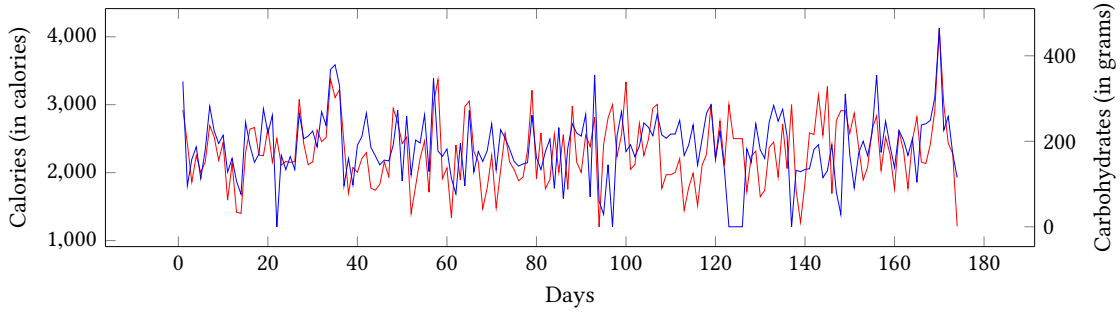


Fig. 8. Calorie (red) and Carbohydrate (blue) Intake Data (superimposed for multivariate analysis).

Calorie and Carbohydrate Intake: Multivariate Summaries. What if there is a correlation between the calorie and carbohydrate intake for this particular user? We can find out by looking at the multivariate summaries. Fig. 8 shows both of them superimposed by day, for our example user. It is hard to discern common trends and patterns directly from the raw multivariate time-series data. In contrast, for the joint calorie and carbohydrate intake data, our system produces 27 individual-level summaries using 11 different protoforms and 18 group-level summaries using 8 different protoforms. Representative individual-level summaries (13 of them) are shown in Table 5 and group-level summaries (10 of them) are shown in Table 6. It seems that our user is performing much better with carbohydrate intake when compared against how well they are performing with their calorie intake. The rest of the users seem to perform well on Mondays.

4.1.2 Heart Rate: Insight4Wear. Fig. 9 shows a snippet of heart rate data for one user that spans over 400 days. We believe that SAX equiprobable bins are actually not ideal for heart rate data. Instead, we generate meaningful

Table 5. Multivariate Individual-Level Summaries for Calorie and Carbohydrate Intake

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
Standard Evaluation (TW)	In the past full week, your calorie intake has been high and your carbohydrate intake has been very high.	N/A	N/A	1	0.98	0.06
Standard Evaluation (sTW)	On some of the days in the past week, your calorie intake has been low and your carbohydrate intake has been high.	0.93	1	0.29	0.23	0.43
Standard Evaluation (sTW) w/ qualifier	On all of the days in the past week when your calorie intake was very low, your carbohydrate intake was moderate.	1	1	1	0.95	0.43
Standard Evaluation + Goal	On some of the days in the past week, you did not reach your goal to keep your calorie intake low and you reached your goal to keep your carbohydrate intake low.	N/A	N/A	0.43	0.71	0.26
Comparison	Your calorie intake was higher and your carbohydrate intake was higher than they were the week before.	N/A	N/A	1	0	1
Comparison + Goal	You did not do as well overall with keeping your calorie intake low and you did not do as well overall with keeping your carbohydrate intake low than you did the week before.	N/A	N/A	1	0	1
Standard Trend	Some of the time, your calorie intake increases and your carbohydrate intake increases from one day to the next.	1	1	0.32	0.06	1
Cluster-Based Pattern	This past week, your calorie intake was moderate, then very low, then high, then very high, then low, then moderate and your carbohydrate intake was moderate, then high, then very low, then high. During half of the weeks similar to this past one, your calorie intake rose and your carbohydrate intake dropped the next week.	1	1	0.5	0.38	0.47
Standard Pattern	The last time you had a week like this past one, your calorie intake rose and your carbohydrate intake stayed the same the next week.	N/A	N/A	1	0.93	0.93
If-Then Pattern	There is 100% confidence that, when your calorie intake follows the pattern of being very high, then very high, your carbohydrate intake tends to be very high the next day.	N/A	N/A	1	0.23	0.24
Day If-Then Pattern	There is 100% confidence that, when your calorie intake follows the pattern of being very high on a Saturday, your calorie intake tends to be very high the next Sunday and your carbohydrate intake tends to be very high the next Sunday.	N/A	N/A	1	0.17	0.2
Day-Based Pattern	Your calorie intake tends to be low and your carbohydrate intake tends to be low on Mondays.	1	1	0.04	0.01	0.01
Goal Assistance	In order to better to follow the 2000-calorie diet, you should decrease your calorie intake and increase your carbohydrate intake.	N/A	N/A	N/A	N/A	N/A

ranges for heart rate; these are shown and contrasted with the SAX symbols in Table 7. As can be seen, a different set of summarizers is used as well. We believe that the SAX binning is not ideal because heart rate data has very little temporal variation. Looking at the data, the data points are strictly between 60 and 100 beats per minute (bpm). Due to the lack of temporal variation, the SAX representations for each granularity will be heavily affected. The letters chosen for each day or week will make data points seem to differ greatly when, in reality, the differences are minimal. For instance, the standard evaluation summaries at the daily and weekly granularities

Table 6. Multivariate Group-Level Summaries for Calorie and Carbohydrate Intake

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
Standard Evaluation (TW)	Almost none of the users in this study had a very low calorie intake and a very low carbohydrate intake in the past full week.	1	0.96	0.04	0	0.1
Standard Evaluation (sTW)	Almost none of the users in this study had a low calorie intake and a moderate carbohydrate intake on some of the days in the past week.	1	1	0.04	0	0
Standard Evaluation (sTW) w/ qualifier	Some of the users in this study had a very low carbohydrate intake when they had a very low calorie intake on all of the days in the past week.	0.95	1	0.29	0	1
Standard Evaluation + Goal	Some of the users in this study reached their goal to keep their calorie intake low and did not reach their goal to keep their carbohydrate intake low on all of the days in the past week.	0.98	0.97	0.3	0	1
Comparison	Some of the users in this study had a lower calorie intake and a lower carbohydrate intake than they did the week before.	0.8	0.92	0.26	0	1
Comparison + Goal	Some of the users in this study did better with keeping their calorie intake low and better with keeping their carbohydrate intake low than they did the week before.	0.8	0.92	0.26	0	1
Standard Trend	Most of the users in this study increase their calorie intake and increase their carbohydrate intake from one day to the next some of the time.	1	1	0.81	0	1
Cluster-Based Pattern	After looking at clusters containing weeks similar to this past one, it can be seen that almost none of the users with these clusters may see a drop in their calorie intake and a rise in their carbohydrate intake next week.	1	0.91	0.03	0	0.03
Standard Pattern	Based on the most recent weeks similar to this past one, it can be seen that almost none of the users may see a drop in their calorie intake and a rise in their carbohydrate intake next week.	1	0.91	0.04	0	0.04
General If-Then Pattern	For most of the users in this study, it is true that when they had a very low carbohydrate intake, they had a very low calorie intake.	1	0.79	0.78	0	1
Day-Based Pattern	More than half of the users in this study tend to have a very low calorie intake and a very low carbohydrate intake on Mondays.	0.96	1	0.6	0	1
Goal Assistance	More than half of the users in this study have been given advice to increase their calorie intake.	1	0.93	0.7	0	1

both state the user's average daily heart rate to be "low," even though the heart rate is within a healthy range and only differs slightly from other data points. As heart rate is more about staying within a healthy range, it is better to create our own discretization for this particular dataset.

Our system produces 21 summaries using 9 different protoforms, with representative summaries shown in Table 8. Unlike the user for the calorie intake study, this user does not have trouble satisfying the goal of keeping their heart rate within range. For example, the if-then pattern summary suggests that, whenever the user has six

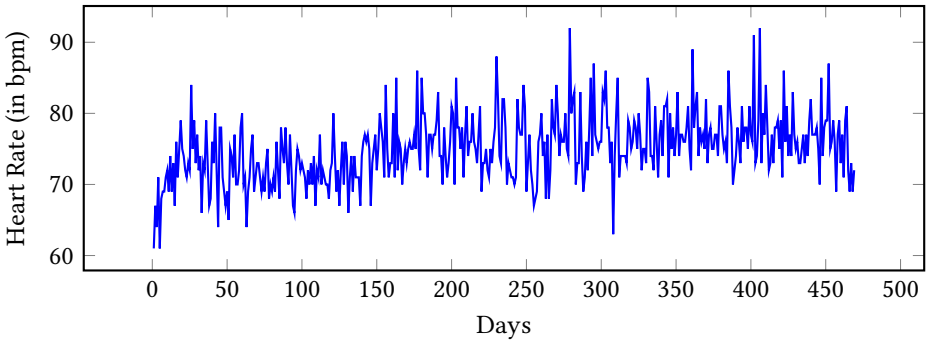


Fig. 9. Heart Rate Data Snippet (Insight4Wear)

Table 7. Mapping of Heart Rate Data Using SAX and Meaningful Ranges

SAX		Meaningful Ranges	
Symbol	Summarizer	Value Range	Summarizer
a	very low	0 - 50	abnormally low
b	low	50 - 60	low
c	moderate	60 - 110	within range
d	high	110 - 120	high
e	very high	120 and up	abnormally high

Table 8. Summaries for Average Daily Heart Rate

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
`Standard Evaluation (TW)`	In the past full week, your heart rate has been within range.	N/A	N/A	1	0	1
`Standard Evaluation (sTW)`	On all of the days in the past week, your heart rate has been within range.	1	1	1	0	1
`Standard Evaluation (sTW) + Goal`	On all of the days in the past week, you reached your goal to keep your heart rate within range.	1	1	1	0	1
`Comparison`	Your heart rate was lower than it was the week before.	N/A	N/A	1	0	1
`Comparison + Goal`	You did about the same overall with keeping your heart rate within range than you did the week before.	N/A	N/A	1	0	1
`Standard Trend`	Half of the time, your heart rate increases from one day to the next.	0.56	0.74	0.46	0.54	1
`If-Then Pattern`	There is 100% confidence that, when your heart rate follows the pattern of being within range, your heart rate tends to be within range the next day.	N/A	N/A	0.46	0	1
`Day If-Then Pattern`	There is 100% confidence that, when your heart rate follows the pattern of being within range on a Saturday, your heart rate tends to be within range the next Sunday.	N/A	N/A	0.46	0.33	0.67
`Day-Based Pattern`	Your heart rate tends to be within range on Wednesdays.	1	1	1	0	1

days of within range behavior, the heart rate on the following day also remains within range. This study also showcases the power of our framework, since we can generate meaningful summaries by simply changing the symbolic mappings and adjusting the summarizers.

4.2 Trying Different Parameters

In this section, we will explore different parameter values when running our system on our user's calorie intake data. We do not show the summaries generated using different values, but representative summaries can be found in the supplementary materials[1].

4.2.1 Time Window. We tried different time windows to use in order to look for more patterns. What if our user wished to look at months instead of weeks? How about the entire time frame? For our earlier experiments, we used a weekly time window with a daily sub-time window. We re-ran our experiments for a monthly time window with a daily sub-time window, and for no time window (where the entire time frame is evaluated).

When we switched to the monthly granularity, the system produced 17 individual-level summaries using 6 protoforms and 13 group-level summaries using 5 protoforms. The change in time window affects every summary type except the standard trend and the day-based pattern summaries since they do not depend on the input time window. The output also does not contain normal if-then pattern summaries. This is not very surprising since the calorie intake data contains only six months of data (174 days), and thus there is not enough data to extract meaningful or frequent monthly patterns. The results are also very different with the group-level summaries, although the same summary types are present since the set of group-level types is derived from the set of individual-level summary types. As for the summaries themselves, we can see a difference in conclusion between the standard evaluation (sTW) summaries at the weekly and monthly granularities. We can see that the days in the past week were not very representative of the calorie intake for the entire month.

When we remove the time window, all summaries evaluate the entire time frame. The system produces 10 summaries using 3 protoforms for both individual-level and group-level summary output. The only summary types that work without a time window are the standard evaluation (sTW), goal evaluation, standard trend, and day-based pattern types. These summary types can be used for the entire dataset where no time window needs to be specified. Similar outcomes are observed for group-level summaries.

4.2.2 Alphabet Size. The alphabet size determines the number of letters we use to discretize the time-series data. The chosen default alphabet size is 5, which allows our framework to use letters "a" through "e" in the alphabet. What if our user wanted their summaries to be more/less precise? When we change the alphabet size, we may be able to find different patterns as the data points will be assigned different letters. Additionally, we will have different sets of summarizers for the summaries we generate. We re-ran our experiments and display the number of individual and group level summaries across alphabet sizes 3, 5, and 7 in Table 9.

Table 9. Number of Summaries Generated per Alphabet Size

Alphabet Size	Individual-Level	Group-Level
3	22	16
5	19	16
7	17	16

Overall, we can see a decrease in the number of individual-level summaries whenever the alphabet size increases. This may reflect a decrease in the number of if-then pattern summaries found as more letters are used to create the SAX representation.

[1]See "ACM Supplementary Materials.pdf" at https://github.com/harrij15/TemporalSummaries

4.2.3 Minimum Support and Confidence Thresholds. These thresholds mainly control the output of the if-then pattern summaries. The default thresholds are 20% for minimum support and 80% for minimum threshold. What if our user wanted patterns that occurred more or less frequently? We re-ran our experiments for different minimum support and confidence thresholds (e.g., 20%, 50%, or 80%). We did not find any if-then patterns for minimum support of 50% an 80%, and thus we show results for the lower support threshold. Note that minimum support of 0 means that we consider all patterns that occur one or more times in the data. As we can see from the results in Table 10, all 118 of the frequent patterns found for our user's calorie intake data occur less than half of the time (since no sequences reached the 50% threshold). Only 15 sequences surpass the 20% support threshold while only 5 sequences surpass the 50% confidence threshold. For the calorie intake data in particular, the default thresholds filter out most of the discovered if-then patterns to just three patterns.

Table 10. Minimum Support and Confidence Thresholds

Minimum Support	Minimum Confidence	# of If-Then Pattern Summaries
0	0	118
0	0.2	83
0	0.5	5
0	0.8	5
0.2	0	15
0.2	0.2	15
0.2	0.5	3
0.2	0.8	3

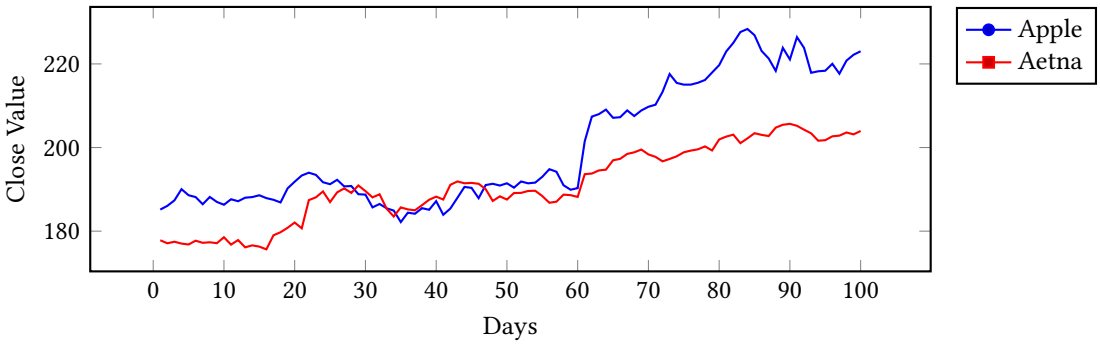


Fig. 10. Close Value Data for Apple and Aetna

4.2.4 State-Of-The-Art Comparison. Although there is a lack of open-source or publicly available automatic natural language summarization systems in the personal health domain, we qualitatively compare our summary output versus other systems. We decided to look at the stock market [2] and weather [26] domains.

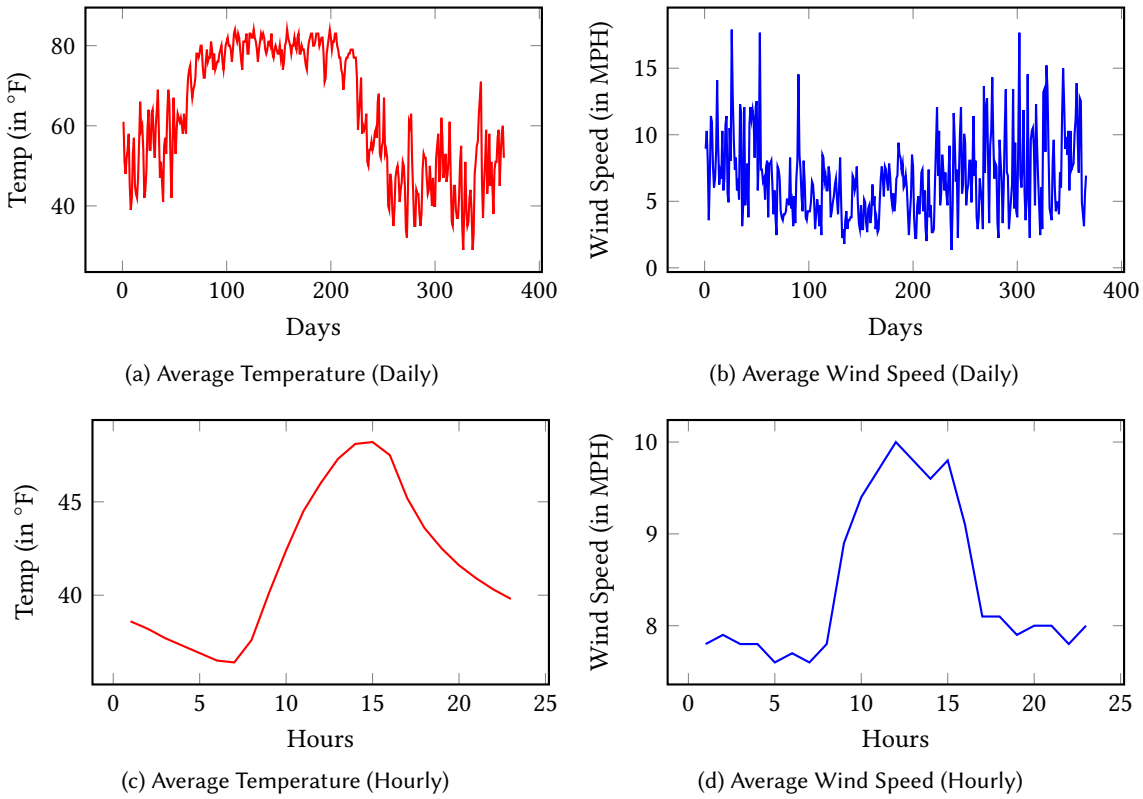
Stock Market Data: In the stock market domain, Aoki et al. [2] extended an encoder-decoder model created by Murakami et al. [29] in order to generate comments about the Nikkei stock market. They generate summaries about the general trend of the stock market time-series ticker data, such as "*Nikkei turns lower as yen's rise hits exporters*" and "*Nikkei Stock Average opens at a high price after Dow Jones Industrial Average closes at a high price.*"

Their main extension is the ability to handle multivariate input. For our work, we can apply our protoform-based approach to stock market data gathered using the REST API from AlphaVantage [36]. With this API, we retrieved a snippet of 100 days of Apple's and Aetna's stock market data beginning from May 2018, as plotted in Fig. 10. Our system is able to provide more insights as shown in Table 11. It generated a total of 242 multivariate summaries, which were slightly modified (protoform-wise) to match the stock market data. We find patterns that cannot be as easily seen and our summaries say a lot more about the data. The protoform-based approach also has better performance in terms of how quickly the summaries are generated.

Table 11. Apple and Aetna - Stock Market Summaries (AlphaVantage)

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
Standard Evaluation (TW)	In the past full week, the AAPL close value has been very high and the AET close value has been very high.	N/A	N/A	1	0.92	1
Standard Evaluation (sTW)	On all of the days in the past week, the AAPL close value has been very high and the AET close value has been very high.	1	1	1	0.92	1
Standard Evaluation (sTW) w/ qualifier	On all of the days in the past week when the AAPL close value was very high, the AET close value was very high.	1	1	1	0.92	1
Comparison	The AAPL close value was about the same and the AET close value was about the same as they were the week before.	N/A	N/A	1	0	1
Standard Trend	Some of the time, the AAPL close value increases and the AET close value increases from one day to the next.	1	1	0.34	0.004	1
Cluster-Based Pattern	This past week, the AAPL close value was very high and the AET close value was very high. During all of the weeks similar to this past one, the AAPL close value stayed the same and the AET close value stayed the same the next week.	1	1	1	0.75	1
Standard Pattern	The last time you had a week like this past one, the AAPL close value stayed the same and the AET close value stayed the same the next week.	N/A	N/A	1	0.75	1
If-Then Pattern	There is 100% confidence that, when the AAPL close value follows the pattern of being high, the AET close value tends to be high, then high the next day.	N/A	N/A	1	0.19	0.2
Day If-Then Pattern	There is 100% confidence that, when the AET close value follows the pattern of being very low on a Thursday, the AAPL close value tends to be low the next Friday and the AET close value tends to be very low the next Friday.	N/A	N/A	1	0.13	0.2
General If-Then Pattern	In general, if the AAPL close value is very low, then the AET close value is very low.	0.55	1	0.45	0.41	0.7
Day-Based Pattern	the AAPL close value tends to be very low and the AET close value tends to be very low on Mondays.	1	1	0.05	0.01	0.7

Weather Data: In the weather domain, the SUMTIME system proposes a general time-series summarization model [26]. They focus their efforts on the weather domain where they describe the forecast of the next 12-24 hours in natural language. An example summary is "W 8-13 backing SW by mid afternoon and S 10-15 by midnight," which describes wind direction and speed [34]. Using our framework, we generated summaries describing the average temperature and the average wind speed tracked by the weather station at the Huntsville International Airport in Huntsville, Alabama. This data was provided by the National Centers For Environmental Information (NCEI) [27]. We used two datasets, one containing a year of daily data between March 1, 2018 and March 1, 2019 and the other containing a day of hourly data for January 1, 2010. We display figures for temperature and wind speed daily data in Figures 11a and 11b. Our framework generates 52 summaries at the weekly (TW) granularity, some of which can be found in Table 12.



(a) Average Temperature (Daily)

(b) Average Wind Speed (Daily)

(c) Average Temperature (Hourly)

(d) Average Wind Speed (Hourly)

Fig. 11. Average daily temperature and wind speed data for Huntsville, AL (for one year and for one day)

For the hourly data, we display average temperature and wind speed data in Figures 11c and 11d. We generate 11 summaries in total with the multivariate summaries shown below. We can see that some of the summary types do not show as we stick to mainly sub-time window conclusions. Some of the protoforms were also modified to account for the change in granularity.

5 RELATED WORK

Data-to-text generation methods include statistical [24] and neural [23] machine translation, and rule-based linguistic summarization methods [6]. van der Lee et al. [38] compared the performance and text quality between rule-based, neural, and statistical methods, concluding that rule-based methods generally perform faster and produce higher text quality, although the manual creation of the sentence prototypes is time-intensive. Since text quality is important in personal health, we follow the rule-based linguistic paradigm.

Linguistic database summarization methods rely on the concept of protoforms and fuzzy logic [6, 44] to summarize data. Linguistic summarization of time-series data has been used for various applications, such as elderly care [40], physical activity tracking [33], driving simulation environments [13], deforestation analysis [11], human gait study [1], periodicity detection [28], time-series forecasting [22], and generation of longer temporal "narratives" from neonatal intensive care data via the use of a neonatal ontology [14]. Other work includes the use of genetic algorithms [7] to generate linguistic summaries from time-series, and those that place emphasis

Table 12. Huntsville, Alabama - Temperature and Wind Speed Summaries (NCEI)

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
Standard Evaluation (TW)	In the past full week, the average temperature has been low and the average wind speed has been moderate.	N/A	N/A	1	0.95	0.66
Standard Evaluation (sTW)	On some of the days in the past week, the average temperature has been low and the average wind speed has been very low.	0.93	1	0.29	0.24	0
Standard Evaluation (sTW) w/ qualifier	On all of the days in the past week when the average temperature was very low, the average wind speed was low.	1	1	1	0.94	0.1
Comparison	The average temperature was higher and the average wind speed was lower in week 51 than they were in week 50.	N/A	N/A	0.5	0.5	0.66
Standard Trend	Some of the time, the average temperature increases and the average wind speed increases from one day to the next.	0.88	0.94	0.28	0.04	1
If-Then Pattern	There is 100% confidence that, when the average temperature follows the pattern of being very low, the average temperature tends to be very low and the average wind speed tends to be very high the next day.	N/A	N/A	0.28	0.17	0.21
Day-Based Pattern	The average temperature tends to be very low and the average wind speed tends to be very low on Wednesdays.	1	1	0.06	0.01	0.1

Table 13. Huntsville, Alabama - Hourly Temperature and Wind Speed Summaries (NCEI)

Protoform Type	Summary	T_1	T_2	T_3	T_4	T_5
Standard Evaluation (sTW)	During almost none of the hours in the past day, the average temperature was very low and the average wind speed was very low.	1	1	0.13	0.09	0.95
Standard Evaluation (sTW) w/ qualifier	During all of the hours in the past day when the average wind speed was very low, the average temperature was very low.	1	1	1	0.94	0.1
Comparison	The average temperature was about the same and the average wind speed was about the same in hour 22 as they were in hour 21.	N/A	N/A	0.5	0.5	0.66
Standard Trend	During some of the day, the average temperature decreases and the average wind speed decreases from one hour to the next.	0.88	0.94	0.28	0.04	1

on simple trends (e.g., increasing, concave) [20]. *In contrast to these works, we propose a more comprehensive set of summaries, and unlike all previous time-series summary-based works, we also apply data mining to discover interesting patterns across multiple variables to produce more interesting explanations.* [2]

There are many works on time-series data mining, reviewed by Batyrshin and Sheremetov [5], including the construction of rules based on patterns found in the data [12], using derivatives to describe the concavity/convexity of trends [8], identification of pre-determined patterns using shape descriptors [4], transformation of time series into state intervals to create association rules [18], generating reports about stocks [32], and so on. *These approaches find temporal patterns or rules based on shapes and trends, but they do not generate explanations as we do via our temporal summaries.* The closest work to ours is [16], where they generate linguistic descriptions of multivariate data via feature extraction, primitive pattern extraction via neural networks, and rule generation [37]. In their work, they find repeated patterns of events in time series and generate natural language describing each event in a sequence. They also use a rule generation algorithm called *sig** that allows them to find temporal rules to identify sleep apnea in patients. An example summary of an event from their work would be "no airflow and

[2]While there is work on summaries spanning multiple variables in the context of neonatal intensive care data [14], that work is based on a neonatal ontology and does not do multivariate temporal pattern mining as in our work.

no chest and abdomen wall movements without snoring." In our work, we mine frequent sequences to generate more interesting if-then summaries, as well as cluster-based summaries. Our framework is also able to provide many more informative summaries based on the comprehensive set of univariate and multivariate protoforms. The natural language in the summaries we generate also sound more natural.

Recent work on time-series summaries includes [29] that uses an encoder-decoder model to generate natural language summaries in the financial domain, and [2] that extends the model to multiple external factors (e.g., relationships between the Nikkei and Dow Jones stock market data). In their training data, they paired up a time series with a market comment that aligns with it. They used 16,276 headlines gathered from the Nikkei Quick News (NQN) to train their model. These headlines spanned from December 2010 to October 2015. Aoki et al. [2] used five-minute charts of seven stock market indices from Thomson Reuters DataScope Select[3]. However, these summaries are limited to simpler conclusions (e.g., a continual rising trend). As such, neural network based methods suffer from several drawbacks, such as lack of high quality summaries [38], dependence on large training data and/or supervision, and lack of ability to explain patterns directly from raw temporal data. *In contrast, our system is unsupervised and generates summaries that explain interesting patterns and trends that are not immediately apparent (based on pattern mining and clustering).*

6 CONCLUSION

We presented a system to automatically generate explanations from a user's personal health data. Unlike most previous approaches that either focus on tabular, textual, or relatively simple trend summaries, we mine interesting patterns from symbolic representations of numeric temporal data, and propose a comprehensive set of useful summaries that cover a wide range of scenarios. We showcase our work using real user data. Our system is designed to extract comprehensible summaries to better guide users towards their goals. To our knowledge, this is the first, comprehensive and systematic approach to generate explainable summaries from time-series personal health data via protoforms. There is *no current system* that can automatically extract patterns and clusters from time-series data and present them to the user in an explainable manner in natural language. In fact, our approach is also *generic* and *extensible* to other domains, and not just health-related data.

It is important to note that our main contribution in this paper is the comprehensive framework for the generation of useful and informative explanations of time-series data. In the future, we also aim to analyze how our summaries ultimately impact the behavior of users via a user study. After conducting this study, we hope to further improve our summary output by modifying our protoform hierarchy. We will focus on which findings are actually the most interesting and helpful to users, along with the most comprehensible ways to put these findings into words.

On top of this, we also seek to fully automate the summarization process where the use of protoforms are no longer needed, while retaining the efficiency and readability of our summaries. Our next step will include the incorporation of time-series shapes where we can both use these shapes to extend our current framework and to attempt further automation of summarization. With these shapes, we will be able to better summarize a time series by describing where interesting shapes (e..g, certain spikes or drops) occur within and across time series. These descriptions could be seen as creating a narrative about a time series within a specific window when applied to the personal health domain. We also aim to use deep learning for and over the shape alphabet to automatically generate summaries based on what shapes we can find in a time series. Finally, we seek to incorporate more complex summaries that enable goal assistance, and those that highlight clusters, anomalies, and other interesting patterns in the data.

[3]https://hosted.datascope.reuters.com/DataScope/

ACKNOWLEDGMENTS

This work is supported by IBM Research AI through the AI Horizons Network, and was conducted under the auspices of the RPI-IBM Health Empowerment through Analytics Learning and Semantics (HEALS) project.

REFERENCES

[1] Alberto Alvarez-Alvarez and Gracian Trivino. 2013. Linguistic description of the human gait quality. *Engineering Applications of Artificial Intelligence* 26, 1 (2013), 13 – 23.
[2] Tatsuya Aoki, Akira Miyazawa, Tatsuya Ishigaki, Keiichi Goshima, Kasumi Aoki, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2018. Generating Market Comments Referring to External Resources. In *INLG Conf.*
[3] American Diabetes Association et al. 2019. 5. Lifestyle management: standards of medical care in diabetesâĂŤ2019. *Diabetes Care* 42, Supplement 1 (2019), S46–S60.
[4] James Baldwin, T.P. Martin, and J.M. Rossiter. 1998. Time series modelling and prediction using fuzzy trend information. *Int'l Conf. on Soft Computing and Information Intelligent Systems* (1998).
[5] I.Z. Batyrshin and L.B. Sheremetov. 2008. Perception-based approach to time series data mining. *Applied Soft Computing* 8, 3 (2008), 1211 – 1221. Forging the Frontiers – Soft Computing.
[6] Fatih Emre Boran, Diyar Akay, and Ronald R Yager. 2016. An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications* 61 (2016), 356–377.
[7] Rita Castillo-Ortega, Nicolás Marín, Daniel Sánchez, and Andrea Tettamanzi. 2011. Linguistic Summarization of Time Series Data using Genetic Algorithms. In *Conf. of the European Society for Fuzzy Logic and Technology*.
[8] J.T.-Y. Cheung and G. Stephanopoulos. 1990. Representation of process trends – Part I. A formal representation framework. *Computers & Chemical Engineering* 14, 4 (1990), 495 – 510.
[9] Eun Kyoung Choe, Nicole B. Lee, Bongshin Lee, Wanda Pratt, and Julie A. Kientz. 2014. Understanding quantified-selfers' practices in collecting and exploring personal data. In *ACM conf. on Human Factors in Computing Systems*.
[10] J. Codella, C. Partovian, H.-Y. Chang, and C. H. Chen. 2018. Data quality challenges for person-generated health and wellness data. *IBM Journal of Research and Development* 62, 1 (Jan 2018), 3:1–3:8.
[11] Patricia Conde-Clemente, Jose M. Alonso, ÃĽldman O. Nunes, Angel Sanchez, and Gracian Trivino. 2017. New types of computational perceptions: Linguistic descriptions in deforestation analysis. *Expert Systems with Applications* 85 (2017), 46 – 60.
[12] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. 1998. Rule Discovery from Time Series. In *SIGKDD Conf.*
[13] Luka Eciolaza, MartÃŋn Pereira-FariÃśa, and Gracian Trivino. 2013. Automatic linguistic reporting in driving simulation environments. *Applied Soft Computing* 13, 9 (2013), 3956 – 3967. http://www.sciencedirect.com/science/article/pii/S1568494612004231
[14] Albert Gatt, François Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. From Data to Text in the Neonatal Intensive Care Unit: Using NLG Technology for Decision Support and Information Management. *AI Commun.* 22, 3 (Aug. 2009), 153–186.
[15] Herbert Paul Grice. 1967. Logic and Conversation. In *Studies in the Way of Words*, Paul Grice (Ed.). Harvard University Press, 41–58.
[16] Gabriela Guimarães and Alfred Ultsch. 1999. A Method for Temporal Knowledge Conversion. In *Advances in Intelligent Data Analysis*, David J. Hand, Joost N. Kok, and Michael R. Berthold (Eds.). 369–380.
[17] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2002. Squeezer: An efficient algorithm for clustering categorical data. *Journal of Computer Science and Technology* 17 (09 2002), 611–624.
[18] Frank Höppner. 2001. Learning Temporal Rules from State Sequences. In *IJCAI Workshop on Learning from Temporal and Spatial Data*.
[19] A. Jain and J. M. Keller. 2015. Textual summarization of events leading to health alerts. In *International Conf. of IEEE Engineering in Medicine and Biology Society*.
[20] Janusz Kacprzyk, Anna Wilbik, and Slawomir Zadrożny. 2010. An Approach to the Linguistic Summarization of Time Series Using a Fuzzy Quantifier Driven Aggregation. *Int. J. Intell. Syst.* 25, 5 (May 2010), 411–439.
[21] Janusz Kacprzyk, Ronald R Yager, and Slawomir Zadrozny. 2002. Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support. In *Knowledge discovery for business info. systems*.
[22] Katarzyna Kaczmarek-Majer and Olgierd Hryniewicz. 2019. Application of linguistic summarization methods in time series forecasting. *Information Sciences* 478 (2019), 580 – 594.
[23] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *CoRR* abs/1701.02810 (2017). arXiv:1701.02810
[24] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL Companion Volume: Demo and Poster Sessions*.

[25] Jessica Lin, Eamonn J. Keogh, Li Wei, and Stefano Lonardi. 2007. Experiencing SAX: A Novel Symbolic Representation of Time Series. *Data Min. Knowl. Discov.* 15 (08 2007), 107–144.
[26] Walter Maner and Sean Joyce. 1997. WXSYS Weather Lore + Fuzzy Logic = Weather Forecasts. (01 1997). https://www.researchgate.net/publication/237546595_WXSYS_Weather_Lore_Fuzzy_Logic_Weather_Forecasts
[27] Matthew J. Menne, Imke Durre, Bryant Korzeniewski, Shelley McNeal, Kristy Thomas, Xungang Yin, Steven Anthony, Ron Ray, Russell S. Vose, Byron E.Gleason, and Tamara G. Houston. 2020. Global Historical Climatology Network - Daily (GHCN-Daily), Version 3. https://www.ncei.noaa.gov/
[28] Gilles Moyse and Marie-Jeanne Lesot. 2016. Linguistic summaries of locally periodic time series. *Fuzzy Sets and Systems* 285 (2016), 94 – 117.
[29] Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. 2017. Learning to Generate Market Comments from Stock Prices. In *ACM Annual Meeting*.
[30] Elizabeth Peel, Margaret Douglas, and Julia Lawton. 2007. Self monitoring of blood glucose in type 2 diabetes: longitudinal qualitative study of patients' perspectives. *BMJ* 335, 7618 (Sep 2007), 493.
[31] Reza Rawassizadeh, Elaheh Momeni, Chelsea Dobbins, Joobin Gharibshah, and Michael Pazzani. 2016. Scalable Daily Human Behavioral Pattern Mining from Multivariate Temporal Data. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (Nov. 2016), 3098–3112.
[32] Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems.* Cambridge University Press.
[33] Daniel Sanchez-Valdes, Alberto Alvarez-Alvarez, and Gracian Trivino. 2016. Dynamic linguistic descriptions of time series applied to self-track the physical activity. *Fuzzy Sets and Systems* 285 (2016), 162 – 181.
[34] Somayajulu Sripada, Somayajulu G. Sripada, Ehud Reiter, and Ian Davy. 2003. SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator. https://www.researchgate.net/publication/2888176_SumTime-Mousam_Configurable_marine_weather_forecast_generator
[35] Si Sun and Kaitlin L. Costello. 2018. Designing decision-support technologies for patient-generated data in type 1 diabetes. In *AMIA Annual Proceedings*. 1645âĂŞ1654.
[36] Romel Torres. 2019. Alpha Vantage. https://github.com/RomelTorres/alpha_vantage
[37] A. Ultsch. 1993. Knowledge Extraction from Self-Organizing Neural Networks. In *Information and Classification*.
[38] Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2018. Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods. In *INLG Conf.*
[39] Ingmar Weber and Palakorn Achananuparp. 2016. Insights from Machine-Learned Diet Success Prediction. In *Pacific Symp. on Biocomputing*.
[40] Anna Wilbik, James M. Keller, and Gregory L. Alexander. 2011. Linguistic summarization of sensor data for eldercare. *IEEE Int'l Conf. on Systems, Man, and Cybernetics* (2011).
[41] D. Wu, J. M. Mendel, and J. Joo. 2010. Linguistic summarization using IF-THEN rules. In *Int'l Conf. on Fuzzy Systems*.
[42] Ronald R. Yager. 1982. A new approach to the summarization of data. *Information Sciences* 28, 1 (1982), 69 – 86.
[43] Lotfi A. Zadeh. 1983. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications* 9, 1 (1983), 149 – 184.
[44] Lotfi A. Zadeh. 2002. A prototype-centered approach to adding deduction capability to search engines-the concept of protoform. In *IEEE Symposium on Intelligent Systems*.
[45] Mohammed J. Zaki. 2001. SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning* 42, 1 (01 Jan 2001), 31–60.