TEXTUAL SUMMARIZATION OF DATA USING LINGUISTIC PROTOFORM SUMMARIES

A Thesis Presented to the Faculty of Graduate School
University of Missouri – Columbia

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

by
AKSHAY JAIN
Dr. James M. Keller, Thesis Supervisor
MAY 2015
The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

**TEXTUAL SUMMARIZATION OF DATA USING LINGUISTIC PROTOFORM SUMMARIES**

presented by Akshay Jain

a candidate for the degree of Master of Science

and hereby certify that, in their opinion, it is worthy of acceptance.

______________________________

Dr. James M Keller

______________________________

Dr. Marjorie Skubic

______________________________

Dr. Mihail Popescu
ACKNOWLEDGEMENTS

I would like to thank my adviser Dr. James Keller for his patience and invaluable guidance throughout this work. He continues to be a source of inspiration and I am glad to further my graduate studies under his supervision. I am very grateful to Dr. Marjorie Skubic for providing me financial support and encouragement, without which this work would not have been possible. I am also thankful to her for providing an interdisciplinary environment which helps me realize the importance of this work. I am appreciative of Dr. James Bezdek and Dr. Mihail Popescu for providing ideas related to this thesis. I thank Dr. Popescu also for agreeing to be in my thesis committee.

I am indebted to Dr. Guilherme DeSouza for allowing me to work in his lab when I joined the University of Missouri and for the knowledge and experience I gained while being there. Finally, I would like to thank Dr. Scott Frey for granting me a job as a programmer in his lab before I started this work.
# Table of Contents

ACKNOWLEDGEMENTS.................................................................................................................. ii

List of Figures................................................................................................................................... v

List of Tables...................................................................................................................................... vii

ABSTRACT......................................................................................................................................... ix

Chapter 1

Introduction .......................................................................................................................................... 1
  1.1 Problem Statement ...................................................................................................................... 1

  1.2 Overview .................................................................................................................................... 2

  References ......................................................................................................................................... 6

Chapter 2: Paper 1

On the Computation of Semantically Ordered Truth Values of Linguistic Protoform Summaries................................................................. 9

  I. INTRODUCTION ................................................................................................................................. 9

  II. BACKGROUND .................................................................................................................................. 13
      A. Semantically Ordered Quantifiers ................................................................................................. 14

      B. Truth value computation using Zadeh’s calculus of quantified propositions [24]............... 15

      C. Truth value computation using method presented in [2]........................................................ 17

  III. PROPOSED METHOD .................................................................................................................... 20
      A. Lemma 1 ..................................................................................................................................... 21

      B. Theorem 1 .................................................................................................................................. 22

  IV. CASE OF EXTENDED PROTOFORMS .......................................................................................... 24
      A. Truth value computation using Zadeh calculus of linguistically quantified propositions .......... 24

      B. Proposed method adapted to the extended protoforms ............................................................. 27

  V. CONCLUSIONS ............................................................................................................................... 30
ACKNOWLEDGMENT

REFERENCES

Chapter 3: Paper 2
Textual Summarization of Events Leading to Health Alerts

I. INTRODUCTION

II. BACKGROUND
   A. System at TigerPlace
   B. Linguistic Protoform Summaries (LPS)

III. METHODOLOGY

IV. CASE STUDIES

V. CONCLUSION

REFERENCES

Chapter 4
On the use of Linguistic Protoform Summaries as features

4.1 Introduction

4.2 Similarity between LPS using method presented in [2]

4.3 Computing similarity between sets of LPS

4.3.1 Similarity between LPS sets as described in [1]

4.3.2 Taking Quantifier and Summarizer semantics into account while computing dissimilarity between LPS and sets of LPS

4.4 Tests with synthetic data

4.4.1 Semantic variation of LPS dissimilarity along with the data

4.4.2 Computing linguistic prototype of synthetic data

4.5 Computing linguistic prototypes of real data

References

Chapter 5
Conclusions and Future Work

References
List of Figures

Chapter 2

1. Membership functions of Quantifiers.......................................................... 15
2. Illustration of how membership functions are split in order to compute truth values............................................................. 19

Chapter 3

1. System Block diagram............................................................................. 38
2. Membership functions of Quantifiers....................................................... 40
3. Membership functions of Summarizers .................................................. 40
4. Overnight bathroom motion for a 2 week period ........................................ 41
5. Case 1: The alert was generated by bathroom motion ............................... 43
6. Case 2: The alert was generated by bed restlessness................................. 44
7. Case 3: The alert was generated by bathroom motion ............................... 45
8. Case 4: The alert was generated by bed restlessness................................. 45

Chapter 4

1. Membership functions of Summarizers representing Size of balls.............. 51
2. Membership functions of Quantifiers for bag of balls example................... 51
3. Illustration showing variation in data used to compare dissimilarity between LPS sets techniques ............................................................. 63
4. Comparison of Old vs New method for dissimilarity between LPS sets ......... 64
5. Data used for comparing the two LPS prototype computation methods. Each row represents a bag of ball with size of each ball color coded....................... 66
6. Density map example................................................................................. 68
7. Active density maps of residents with mental illness across three points in time. Vertical axis depicts time of day and the horizontal axis depicts day of the month. Color areas depict time and movement in apartment, black areas depict time out of the apartment.............................................. 69
8. Membership functions of Summarizers for density maps………………………… 70
9. Membership functions of Quantifiers for density maps…………………………. 71
10. Density maps for resident case # 1 with the linguistic prototypes for the month before, during and after mental illness………………………………………………………… 74
11. Density maps for resident case # 2 with the linguistic prototypes for the month before, during and after mental illness………………………………………………………… 75
12. Density maps for resident case # 3 with the linguistic prototypes for the month before, during and after mental illness………………………………………………………… 76
13. Density maps for resident case # 4 with the linguistic prototypes for the month before, during and after mental illness………………………………………………………… 77
14. Density maps for resident case # 5 with the linguistic prototypes for the month before, during and after mental illness………………………………………………………… 78
List of Tables

Chapter 2

1. Bigness of each ball ................................................................. 16
2. Truth values using [1], of LPS associated with example 1 .............. 17
3. Truth values using [2], of LPS associated with example 1 .............. 19
4. Truth value computation using proposed method, of LPS associated with example 1 ......................................................... 21
5. Bigness and Heaviness of 4 bags of balls ...................................... 26
6. Truth values of summaries of the form Almost none of the balls are big and heavy, Few of the balls are big and heavy etc. computed using the Zadeh calculus of quantified propositions ........................................... 27
7. Truth values and degree of focus of summaries of the form Almost none of the big balls are heavy, Few of the big balls are heavy etc. computed using the Zadeh calculus of quantified propositions ........................................... 27
8. Truth values of summaries of the form Almost none of the balls are big and heavy, Few of the balls are big and heavy etc. computed using proposed method ................................................................. 30
9. Truth values and degree of focus of summaries of the form Almost none of the big balls are heavy, Few of the big balls are heavy etc. computed using proposed method ................................................................. 30

Chapter 3

1. LPS of the form: The bathroom motion tonight is P than/to A of the nights in last 2 weeks, where A and P are the quantifier and summarizer pair in each row... 41
Chapter 4

1. LPS with truth value ................................................................. 50
2. Summaries with non-zero truth values describing size of balls in a bag ........ 52
3. Summaries describing size of balls in a bag with one summary per summarizer .. 53
4. Heuristically defined semantic similarity between quantifiers defined in Figure 8 .............................................................................. 55
5. Three sets of LPS summarizing size of balls in three bags ....................... 56
6. Weights for Quantifiers shown in Figure 2 ................................................ 57
7. Three sets of LPS summarizing size of balls in three bags ....................... 58
8. Heuristically defined semantic dissimilarity between summarizers defined in Figure 1 .............................................................................. 59
9. Linguistic prototypes of data shown in Figure 5 using the method of [1] ........ 66
10. Linguistic prototypes of data shown in Figure 5 using our method .............. 66
11. Heuristically defined semantic similarity between Quantifiers defined in Figure 9 .............................................................................. 70
12. Heuristically defined semantic Weights of Quantifiers in Figure 9 .............. 70
13. Heuristically defined dissimilarity between Summarizers defined in Figure 8 .... 70
14. Summaries of the form The resident made Q P Out of apartment visits where Q and P are quantifier and summarizer respectively ................................. 72
15. Table showing dissimilarity between the linguistic prototypes for the month when the resident did not have mental illness, with the months during and after mental illness .......................................................... 73
Textual Summarization of Data Using Linguistic Protoform Summaries

AKSHAY JAIN

Dr. James M Keller, Thesis Supervisor

THESIS ABSTRACT

With the unseen quantities of data being generated in all walks of life, varying from social media to health domain, introduction of new techniques in order to better understand this information content is imperative. Contrary to the popular methods such as visualization or statistical summarization which requires the user to adapt to the technology, summarization of this information in the language pertaining to the audience, has recently gained a lot of traction. This thesis deals with textual summarization of data using Linguistic Protoform Summaries (LPS). We start by studying the existing techniques present in the literature to produce LPS, propose a new method and demonstrate its robustness with a mathematical proof. The usefulness of LPS is then illustrated with a novel application in the healthcare domain where the textual summaries are tailored to a clinical population. This is followed by an extensive study on the use of LPS as features in order to process data. There, we present our thoughts on the ways to handle LPS as data features and provide reasoning of this choice. We illustrate this with a real data example where we find a prototypical set of days of the activity of people living in an eldercare facility. Throughout this thesis we design various sets of experiments with synthetic data in order to explain the details of techniques presented.
Chapter 1

Introduction

1.1 Problem Statement

In this digital age, information technology is used to monitor almost all aspects of human life, which gives rise to huge amounts of data. This encompasses the domains of social networking, health sensors, natural phenomena and so on. This data is believed to be rich in information content and can be used to make predictions about future events or present the important aspects of the data to people, who in turn can make useful conclusions. In order to make predictions from the existing data, the algorithms are required to learn from available information and take action accordingly. This thesis is concerned with the presentation side of the problem. That is, how to present the data to the user in a manner such that it can be easy to understand. For instance, presenting the current weather or showing how was your activity in the past week on the basis of myriad of wearable sensors that have surfaced of late. Examples concerned with healthcare domain includes presenting summary of health of a person admitted in an ICU or describing a elder resident’s activity living in a ‘smart home’ equipped with different sensors. One of the widely used techniques is to present this data visually. Visualization possesses some benefits like it enables the user to discover trends in the data and present a lot of information in a screen shot. However, it has learning curve, since the visualization style can change with type of data. Moreover, if the amount of presented data is not trivial, the user might have a hard time viewing it on computer screens or the one’s we carry in our pockets. Needless to say, as the dimensionality of the data increases, visualization becomes increasingly difficult. These issues may still persist even if we mine the data to only present information that is
considered important. Another alternative is to compute some statistical parameters that summarize information content and present them instead of the whole dataset. This technique has the benefit of being concise and avoids the user being overwhelmed with data. However, the downside is that if the parameters are not chosen precisely according to the nature of content, important information might be missed. A different technique that has been gaining traction recently is to summarize the content of the data in natural language. Basically, the data is searched for ‘important information’ which is then presented to the user in the language of preference. Of course such systems may not be directly transferrable across populations speaking different languages, the general framework remains the same nevertheless. This thesis deals with textual summarization of data. We first provide an overview of the different techniques available for this task, followed by our contribution, which is in Linguistic Protoform Summaries (LPS).

1.2 Overview

Two of the most widely used techniques for textual summarization of data are Natural Language Generation (NLG) as described in [1] and Linguistic Protoform Summaries (LPS) which are template based natural language sentences. NLG systems [1] basically comprises of various rules to convert data to text along with some pattern recognition techniques to identify and summarize trends. The NLG group at the University of Aberdeen has done tremendous amount of work in development of such systems for a variety of fields. For instance, the SumTime-Mousam project [2, 3] developed a system to automatically generate textual reports for weather forecasts. In [4] they propose a general system to produce natural language summaries of ontologies. They have multiple systems related to the textual summarization of data in health domain. In [5] they automatically produce nursing shift summaries in Neonatal Intensive care units. Another work described in [6] helps to understand the problems faced by Scuba divers though textual summarization of
data which is collected by the computers attached while they dive. Moving away from the health domain, the system developed in [7] produces textual summaries in order to help navigate around a geographical location, while in [8], the Atlas project, they develop a system to describe spatial data, like census, in text. Regarding the pattern recognition aspect of the NLG domain, in [9] the authors present some techniques to choose the content to present in textual summaries for time series data. Some other work done on NLG includes [10, 11] in which the authors present NLG system along with a visualization system to describe patients medical records to health domain experts.

The major contribution of this thesis is the development of techniques to produce Linguistic Protoform Summaries (LPS) along with exploring their use as features of data. LPS are template based textual summaries describing important aspects of the data in natural language. Language inherently has fuzziness associated with it. Therefore, it is natural to use fuzzy logic for the computation of data to text systems. LPS were first introduced by Yager [12] in order to summarize numeric or textual data. They basically consist of a summarizer, describing some attributes of the data, a quantifier conveying the degree of presence of the summarizer and a truth value (varying from 0 to 1) denoting the validity of the summary corresponding to the underlying data. In the simplest form, a linguistic protoform summary is of the form: $Q$ $y$’s are $P$, where $Q$ and $P$ are the quantifier and summarizer, respectively, with each summary accompanied by truth value. For example: 

*Most of the balls in the bag are big @ 0.8*, is an linguistic protoform summary describing the size of balls inside a bag with *Most* being the quantifier, *big* the summarizer and 0.8 is the truth value of the summary. Since the introduction by Yager, a lot of progress has been made in both in the selection of different templates of the summaries and the ways the truth values are calculated. For a brief overview, we list a few of them. The most basic method to compute the truth values of summaries as used by Yager was using the Zadeh Calculus.
of Quantified propositions [13]. The authors in [14] introduced a method using the Sugeno Integral [15] to describe trends in the data using LPS. In [16] they proposed a new type of protoform, called the extended protoform, to present richer information about the data. Along with the truth value, these protoforms are accompanied by another metric called degree of focus. An interesting use of LPS to compare time series was proposed by [17]. They use linguistic summaries to describe local changes in two time series. In [18] the authors introduce a method to compute distance between two LPS and proved that the distance is a metric. This opened the door for using LPS not just for communication purposes but also as features of data. This makes sense intuitively, since LPS are basically a macro representation of data, distance between two or more LPS can be used as a way to know how the underlying data differ from each other. This method was then used in [19] to cluster a list of LPS and find linguistic prototypes to find the typical nature of large data sets linguistically. In that work, the authors compute linguistic prototypes representing a typical night of a resident living in a smart home equipped with various sensors. This work was then taken forward in [20] to find anomalies in data. They first find linguistic prototypes representing a ‘typical night’ when the person was in normal state. Distance between the prototypes and the LPS of the current night was then used to flag unusual behavior.

This thesis comprises of two conference papers followed an ongoing work chapter. In Chapter 2 we present a paper that was recently accepted in 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2015). In that paper we discuss some theoretical aspects of the methods to compute truth values of LPS. There we carefully studied various techniques present in the literature and point out some shortcomings in them. We also illustrate these with the help of synthetic data. We then provide a new method and show that it overcomes the mentioned problems with the help of a
mathematical proof. Once we have a robust method to compute truth values of simple protoforms, we delve into the case of extended protoforms. Extended protoforms describe relationships between multiple attributes of data. They can be exemplified by: *Most of the balls in the bag are big and heavy*, or, *Most of the small balls in the bag are light*. Along with a truth value, each extended protoform summary can be accompanied by another metric called the degree of focus, providing more information about the validity of the summary. We show that our method is better for these cases also, which is again illustrated by some examples with synthetic data.

A paper submitted to the International Conference of the IEEE Engineering and Biological Society (EMBC) 2015 forms the Chapter 3. In this paper we demonstrate the effectiveness of LPS as a tool to communicate information. We use the technique presented in Chapter 2 to produce textual summaries of sensor data obtained from the ‘smart homes’ of elderly. A system already in place produces alerts on detecting ‘anomalies’ in the live data stream generated by the motion sensor monitoring the movement of residents around the apartment and the bed sensor keeping track of the restlessness of the person during the night. Each sensor stream generates a separate alert. We generate textual summaries of the events leading to these alerts. Since the various sensor modalities are monitoring different aspects of the activity of the same resident, they are all believed to be correlated and it is interesting to observe the data from other sensors along with the sensor parameter that produced the alert. To this end, we select three sensor modalities, namely, the bed restlessness, bathroom motion and the motion of the resident inside the apartment, and accompany each alert with the summaries of all three. Since these summaries are addressed to clinical professionals, we modify the linguistic summaries produced by our method with a set of ‘language rules’ to make them sound more intuitive. To test the effectiveness of our method we discuss four case studies and also mention the comments provided when this system was presented to
our collaborators in healthcare domain. Generally, the textual summarization system was found to have a strong potential to help clinicians understand the resident situation.

In Chapter 4 we study the use LPS as data features. Since LPS are macro representations of the underlying data, distance between them can be used to determine how the information between the data differ. We start by describing the method of [18] and [19] in order to compute dissimilarity between individual LPS and sets of LPS. We then point out at some of the shortcomings of these methods and propose solutions that have a potential to overcome them. We demonstrate these discrepancies with the help of some carefully crafted synthetic data and show that our method helps alleviate these problems. Basically, we reason that the language elements comprising the summaries should follow the associated semantics. We end this discussion by computing linguistic prototypes of data obtained from the homes of elderly which are equipped with sensors. Three one month span of time periods are considered for five residents, each with a month when the person was suffering from mental illness, a month prior to that when they were in ‘normal’ state and a month after the treatment. We observe that the linguistic prototypes present similar information as suggested by visualizing this data.

Finally, in Chapter 5, we lay out the major conclusions of this thesis followed by some directions on future work.

References


Chapter 2

On the Computation of Semantically Ordered Truth Values of Linguistic Protoform Summaries

Akshay Jain*, Student Member, IEEE, and James M. Keller†, Life Fellow, IEEE
Electrical and Computer Engineering
University of Missouri
Columbia, MO, USA
*aj4g2@mail.missouri.edu, †kellerj@missouri.edu

Abstract—Linguistic summaries provide a promise for extracting useful information from large datasets and present them in the form of natural language. Their applications can be found in various disciplines such as eldercare, financial time series etc. In this work we focus on linguistic protoform summaries and point out at some problems associated with the truth value computation methods present in literature. We develop a technique which produces more intuitive truth values for three different kinds of linguistic protoform summaries and illustrate this with help of some examples. Moreover, we show through a mathematical proof that our method is very robust and the truth values always follow the semantic order of the language they are representing.

I. INTRODUCTION

As the amount of data increases, converting it into useful information becomes an increasing difficult problem. This has been widely discussed in the data science community under the name of big data which is said to be comprised of the four V’s – Volume, Variety, Velocity and Veracity [3]. Each of these V’s pose a unique set of problems in terms of understanding
the information content. For instance, a widely used approach in order to get to know the data well is to visualize it. However, as the variety/dimensionality of the data increases visualization becomes more difficult. Another technique is to compute some statistical metrics to get a big picture of the nature of the data. Though the statistical metrics like mean, median etc. provides some insight into the information content, they might neglect a lot of intrinsic details that may be very important. Moreover, it may be difficult for an untrained eye to analyze these metrics to keep up with the data. Linguistic summaries offer a human-centric method to compress and explain large amounts of raw data. For example, [4] discuss the implications of vast amounts of data generated in Neonatal Intensive Care Units and how textual summaries can help the doctors, nurses and caretakers of infants make better use of it. Linguistic summaries hold a promise of providing a solution to these problems by describing the information content in human readable natural language. Along with the capability of describing intricate information in heterogeneous records, it naturally makes it easy for the people with different background to analyze it, for example in case of monitoring sensor output from the home of elderly residents [5].

A number of ways to generate textual summaries of numeric data have been presented in the past. Two prominent approaches are the statistical rule based methods [4, 6] and fuzzy logic techniques [1, 2, 5, 7-13]. In [14] the authors point that statistical rule based techniques produce more sophisticated and longer text summaries of data, while are not so flexible in data processing due to the hard coding of the data selection rules as compared to the fuzzy logic techniques.

Largely, three types of Linguistic Protoform Summaries (LPS) have been talked about in the literature, with each focusing on a different property of the data. They can be exemplified by Many balls are big, Some balls are light and small and Few of the red balls are small.
We call these simple protoform, protoform 2 and protoform 3 summaries, respectively. For each summary one or more metrics are computed indicating its validity with respect to the underlying data. In this work we focus on computation of the metric associated with these three types of summaries.

The summaries of the form Many balls are big were introduced by Yager [1] as a way to summarize both numeric and non-numeric data. Yager’s LPS (and our work) basically is comprised of three terms: a summarizer (e.g., big, small, red etc), a quantifier (e.g., Few, Some, Many etc) and a truth value. Since then this idea has been taken forward both in terms of new protoforms and the way the truth value of a summary is computed. The quantifier forms an integral part of the summaries describing the presence of one or more attributes of data. Historically, a lot of focus has been put on evaluating the sentences based on the type of quantifier used. Generally, they can be used to describe precise information (such as all, none etc.) or can be of imprecise nature (such as Few, Some etc.). The use of fuzzy sets in order to model quantifiers conveying non-crisp information seems to be adequate. Fuzzy quantification is a widely studied topic with many approaches to evaluate the summaries involving fuzzy quantifiers. Authors in [15] provide an in depth analysis of the different quantification techniques. This work builds on a methodology using the Sugeno integral [16]. In the following we present an overview of the development of different aspects of LPS.

In [10] the authors modified the structure of protoforms to characterize different aspects of time series data, namely, how long certain trends remain constant, and how much and how fast they change. They fused the information using the Sugeno Integral [15] to compute truth values of these protoforms. Then in [9], they introduced a metric called degree of focus to reject summaries which do not add much information. An approach to compare two time series was proposed in [8, 16]. They generated LPS to depict the changes in two
time series at various levels. In [5] motion and restlessness data for an elderly participant is analyzed using LPS. Then in [17] a distance metric for a space of linguistic summaries was developed. This allowed the authors to compute distances between groups of linguistic summaries in the eldercare sensor data. The summaries could then be clustered and the clusters eventually represented by Linguistic Medoid Prototypes [13]. Conceptually, the goal of that work was to utilize LPS as a compact but understandable feature set for heterogeneous data streams, both for the purpose of communication and automatic detection of anomalies [18]. A different take to produce summaries at various granularities was proposed in [19] with a method similar to Yager [1] to compute truth values. Reference [20] focusses on summaries of the form Y’s are P and R, e.g., Employees are well paid and young. They call the metric associated with the summaries as the degree of truth, which indicates the percentage of objects satisfying all of the given attributes. The method to compute the degree of truth is somewhat similar to [10], however their main concern is agreement of different attributes rather than the use of quantifiers. In [21, 22], genetic algorithms are used to search for interesting summaries of data. Summaries with features such as high truth value, compactness and uniqueness are deemed as interesting.

The authors of [2] have shown that the Zadeh calculus for defining truth values, (defined and analyzed in Section B), can produce very non-intuitive values [2]. In that paper, a method to address the main problem was introduced, but issues with respect to a natural interpretation of truth values still remained. We build on it to produce truth values which are more intuitive with respect to the semantic meaning of the quantifiers. The truth value of an LPS represents the amount of information it contains or how well it represents the underlying data. In Section II we discuss few of the techniques to compute truth values that are relevant to our work. Section III presents our approach with focus on its benefits accompanied by a theorem proving the robustness of our method. In Section IV, we extend
the proposed method for summaries with extended protoforms, e.g., *Few of the balls are red and small, Few of the red balls are small.*

II. BACKGROUND

LPS are short natural language sentences whose form is prescribed depending on the application and the nature of data to be summarized. A general simple form of linguistic protoform summary is

\[ A \text{ } y\text{'s are } P \]

for example, *Few balls are big.* Here, \( A \) is the linguistic quantifier which is selected according to the quantity of the objects being summarized, like *most, many, few, some* etc. The variable \( y \) represents the type of the objects, like *balls, days* etc. \( P \) is the summarizer, encapsulating a feature, like *big, small, tall* etc. The truth value \( T \) is calculated for every summary representing its informativeness. There are several methods to compute truth values of which we describe two that are relevant to this work. The linguistic variables \( A \) and \( P \) are fuzzy sets over suitable domains and are represented by membership functions defined as normal, convex L-R fuzzy numbers [23] as shown in equation (1).

\[
A(x) = \begin{cases} 
L\left(\frac{x-a}{b-a}\right) & \text{if } x \in [a, b] \\
1 & \text{if } x \in (b, c) \\
R\left(\frac{d-x}{d-c}\right) & \text{if } x \in [c, d] \\
0 & \text{otherwise}
\end{cases}
\]

Where, \( L(x) \) and \( R(x) \) are monotonically non-decreasing shape functions with; \( L(0) = R(0) = 0, L(1) = R(1) = 1 \). While L-R fuzzy numbers are general, many computational problems use either trapezoid functions or triangle functions (\( b = c \)) where \( L \) and \( R \) are linear. It may be the case that the value of the function is 1 at the left or right endpoint of the domain (for e.g., *Almost None* and *Many*, respectively in Figure 1). In such cases, by definition,
\[ L \left( \frac{x-a}{b-a} \right) = 1, \text{if } x = a = b \text{ and } R \left( \frac{d-x}{d-c} \right) = 1, \text{if } x = c = d . \] Also note that for the case of quantifiers, we only concentrate on the ones defined within the domain [0, 1]. However, there is no such restriction for summarizers.

**A. Semantically Ordered Quantifiers**

Words in languages have semantics associated with them. For instance, given a bag containing a number of balls of different sizes, the sentence *Few balls are big* corresponds to fewer number of big balls in the bag as compared to the sentence *Some balls are big*. In line with this, the membership functions of the linguistic quantifiers (*Few* and *Some*, respectively) should also follow the same semantic order. We define semantically ordered quantifiers to follow this reasoning. To fix ideas, suppose that \( A(x) \) and \( B(x) \) are the membership functions of the quantifiers like *Few* and *Some* in the above two sentences (as defined in equation (2) and (3) respectively).

\[
A(x) = \begin{cases} 
L \left( \frac{x-a}{b-a} \right) & \text{if } x \in [a, b] \\
1 & \text{if } x \in (b, c] \\
R \left( \frac{d-x}{d-c} \right) & \text{if } x \in [c, d) \\
0 & \text{otherwise}
\end{cases} \tag{2}
\]

\[
B(x) = \begin{cases} 
L2 \left( \frac{x-p}{q-p} \right) & \text{if } x \in [p, q] \\
1 & \text{if } x \in [q, r] \\
R2 \left( \frac{s-x}{s-r} \right) & \text{if } x \in [r, s] \\
0 & \text{otherwise}
\end{cases} \tag{3}
\]

Then we call these membership functions semantically ordered if,

\[
A(x) \geq B(x) \ \forall \ x \in [a, b] \text{, and } B(x) \geq A(x) \ \forall \ x \in [r, s]
\]
Primarily, we avoid the cases where the functions overlap in an ambiguous fashion. The membership functions of the quantifiers shown in Figure 1 are semantically ordered in the following order: *Almost None, Few, Some, Many*. Trapezoidal functions are usually denoted as Trap(a,b,c,d).

\[ T(A y's \ are \ P) = A \left( \frac{1}{N} \sum_{i=1}^{N} P(y_i) \right) \]  

\[ (4) \]

where \( A(x) \) and \( P(x) \) are the membership functions of the quantifier and summarizer respectively, \( y_i \) is the \( i^{th} \) object in the data to be summarized, with \( N \) total number of objects.

When the data is relatively crisp, this method produces intuitively correct truth values. However, due to the use of averaging operator, a discrepancy arises in case of fuzzier datasets. We illustrate this with the help of some examples.
Example 1

In this example we consider a scenario in which we would like to produce linguistic summaries of bags containing 10 balls of various sizes. We also assume that a separate process provides the summarizer value (bigness) of each ball varying from 0 to 1. Since the validity of LPS can be very subjective, the data in Table 1 is tailored to highlight the problems mentioned above.

We define four semantically ordered quantifiers as shown in Figure 1 and compute truth values of the four sentences – *Almost None of the balls are big*, *Few balls are big*, *Some balls are big* and *Many balls are big*. Note that the membership functions of these quantifiers are of the form presented in equation (1).

According to intuition, for bag 1, *Many balls are big* seems to be the best choice, while for bag 2 and 3, it is safe to say that out of the four summaries, *Almost None of the balls are big* best represents the data, and it should have the highest truth value. Because of the fuzziness of the data, the summaries which do not characterize the data best may have a truth value greater than 0. However, their truth values are expected to follow the semantic order of the quantifiers. Table 2 displays the truth values produced using equation (4) for the data shown in Table 1. The truth value computed for summaries corresponding to bag 1 seems to be in line with the data. However, this is not the case for bags 2 and 3. First, even though in bag 2 none of the balls are really big, the truth value of *Few balls are big* is calculated as maximum. Second, in bag 3, the truth value of *Some balls are big* is computed to be maximum which does not correspond to the fact that only one ball is really big. As noted in

<table>
<thead>
<tr>
<th>Bag 1</th>
<th>0.9</th>
<th>0.9</th>
<th>0.9</th>
<th>0.7</th>
<th>0.7</th>
<th>0.7</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag 2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Bag 3</td>
<td>0.9</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 1: Bigness of each ball
[2], this discrepancy is due to the averaging of memberships in the formula, which works better with monotonically non-decreasing quantifiers (such as *Many*) as compared to others.

<table>
<thead>
<tr>
<th></th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag 1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.80</td>
</tr>
<tr>
<td>Bag 2</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Bag 3</td>
<td>0.00</td>
<td>0.40</td>
<td>0.60</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### C. Truth value computation using method presented in [2]

The problem associated with averaging of the memberships highlighted above was addressed recently by [2]. The reasoning behind their technique is that while summarizing a set of objects humans reject the objects with low memberships. To this end, they use the discrete form of Sugeno Integral [15] to compute the truth value of LPS of the form *A y’s are P*, as shown in equation (5).

\[
T(A \text{ y’s are } P) = \max_{\alpha} (\alpha \land A(P_{\alpha}))
\]

where, \(\land\) is the minimum operator, 
\[
P_{\alpha} = \left\lfloor \frac{\{y_i \in Y \mid P(y_i) \geq \alpha\}}{N} \right\rfloor
\]

is the proportion of objects whose membership in \(P(x)\) is greater than or equal to \(\alpha\), \(|\cdot|\) denotes the cardinality of a set and \(A(x)\) is a normal, convex and monotonically non-decreasing membership function of the quantifier \(A\). Specifically, equation (5) is the Sugeno fuzzy integral with fuzzy measure defined by \(A(x)\). This formulation satisfies the definition of a fuzzy measure as long as \(A(x)\) is a monotonically non-decreasing function. Hence, as shown, it is not suitable to determine the truth values, say, for propositions *Almost None*, *Few* and *Some* in Figure 1. In the following we explain the procedure to calculate the truth value when the quantifiers are not monotonically non-decreasing.
Suppose we have quantifier \( A \), with membership function \( A(x) \) of the form shown in equation (1). In order to compute the truth value of the summary \( A \text{ y's are } P \), \( A(x) \) is first split into \( A_1(x) \) and \( A_2(x) \).

\[
A(x) \Rightarrow \begin{cases} 
A_1(x) = \begin{cases} 
0 & \text{if } x < a \\
L \left( \frac{x-a}{b-a} \right) & \text{if } x \in [a,b] \\
1 & \text{if } x > b 
\end{cases} \\
A_2(x) = \begin{cases} 
1 & \text{if } x < c \\
R \left( \frac{d-x}{d-c} \right) & \text{if } x \in [c,d] \\
0 & \text{if } x > d 
\end{cases}
\end{cases}
\] (6)

The function \( A_1(x) \) is monotonically non-decreasing while \( A_2(x) \) is a monotonically non-increasing function. In order to use equation (5) to compute truth value of a summary involving \( A_2 \), its complement, \( \overline{A_2}(x) \) is defined as shown in equation (7).

\[
\overline{A}_2(x) = \begin{cases} 
0 & \text{if } x < c \\
R \left( \frac{x-c}{d-c} \right) & \text{if } x \in [c,d] \\
1 & \text{if } x > d 
\end{cases}
\] (7)

Then, the truth value of \( A \text{ y's are } P \) is given as

\[
T(A \text{ y's are } P) = T(A_1 \text{ y's are } P) - T(\overline{A}_2 \text{ y's are } P) 
\] (8)

For example, if the function \( A(x) \) is of form \( \text{Trap}(a,b,c,d) \), then it is split as \( A_1(x) = \text{Trap}(a,b,1,1) \) and \( \overline{A}_2(x) = \text{Trap}(c,d,1,1) \), as illustrated in Figure 2. Note that for quantifiers with monotonic non-increasing membership functions such as \textit{Almost none} in Figure 1, the function \( A_1(x) \) is always 1, hence the truth value of \( A_1 \text{ y's are } P \) in this case is also 1. Therefore, the truth value of \( A \text{ y's are } P \), is computed as

\[
T(A \text{ y's are } P) = 1 - T(\overline{A}_2 \text{ y's are } P)
\]
Also, for monotonic non-decreasing quantifiers like *Many* in Figure 1, the function $A_2(x)$ is always 1. Hence,

$$T(A \ y's \ are \ P) = T(A_1 \ y's \ are \ P).$$

Table 3 shows the truth values of the summaries of the form $A \ y's \ are \ big$ computed by this technique with the quantifiers shown in Figure 1. It can be observed that the problem mentioned with the method of [1] does not exist here, and the truth value computed for *Almost None of the balls are big* and *Few balls are big* are more intuitive; *Almost None* being the one with highest truth value for bag 2 and 3. However, the truth value for *Many balls are big* is greater than that of *Few balls are big* and *Some balls are big* in both of these cases, which is counter intuitive according to the data and the semantic order of the quantifiers. Moreover, in bag 1, even though the truth value of *Many balls are big* is correctly computed to be the highest, that of *Few balls are big* and *Almost None of the balls are big* being greater than *Some balls are big*, is again, not intuitive. We address this problem in the next section.

Table 3: Truth values using [2], of LPS associated with example 1

<table>
<thead>
<tr>
<th></th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag 1</td>
<td>0.10</td>
<td>0.20</td>
<td>0.00</td>
<td>0.70</td>
</tr>
<tr>
<td>Bag 2</td>
<td>0.70</td>
<td>0.10</td>
<td>0.10</td>
<td>0.20</td>
</tr>
<tr>
<td>Bag 3</td>
<td>0.70</td>
<td>0.20</td>
<td>0.00</td>
<td>0.30</td>
</tr>
</tbody>
</table>
III. PROPOSED METHOD

The importance of our approach is based on the fact that only one summary may not be sufficient to completely describe a dataset. For example, in [13] all LPS above a threshold were considered to compute linguistic prototypes from a collection of summaries. In such cases, semantically unordered truth values may result in anomalous output. Hence, we lay stress on a technique in which truth values follows the semantic order of the quantifiers. To this end, we modify the method in [2] to produce more intuitive truth values.

For quantifiers $A$, whose membership function $A(x)$ are of the form defined in equation (1), similar to [2], we split them up into $A_1(x)$ and $A_2(x)$ as shown in equation (6). Then, the truth value of $A$ y’s are $P$ is given by:

$$T(A 	ext{ y’s are } P) = T(A_1 	ext{ y’s are } P) \land T(A_2 	ext{ y’s are } P)$$ (9)

where, $A_1(x)$ and $A_2(x)$ are defined as described in Section C and the truth values are computed using equation (5). Note that $T(A_2 	ext{ y’s are } P)$ is evaluated by defining as shown in equation (7).

Table 4 shows the truth value of the bags of balls in Table 1 using proposed method. For bags 2 and 3, the truth value for Almost None of the balls are big is highest and it varies gradually from Few balls are big to Many balls are big. Similarly, for bag 1, the truth value is highest for Many balls are big and it varies intuitively from Some balls are big to Almost none of the balls are big. This evidence suggests that the proposed method does not suffer
from the discrepancies observed in [1] and [2]. We now show that the truth values computed by this method always follow the semantic order of the quantifiers.

**A. Lemma 1**

For any summarizer, $P$, if there are two monotonic quantifiers $A$ and $B$ such that

$$A(x) \geq B(x) \forall x \in [0,1]$$

Then,

$$T(A\text{'s are } P) \geq T(B\text{'s are } P),$$

where,

$$T_A = T(A\text{'s are } P), T_B = T(B\text{'s are } P)$$

are computed using equation (5).

**Proof:**

For each $\alpha$, we have $A(P_{\alpha}) \geq B(P_{\alpha})$. Hence, for any finite set $\alpha_1 \leq \alpha_2 \leq ... \leq \alpha_N = 1$

$$\alpha_1 \land A(P_{\alpha_1}) \geq \alpha_1 \land B(P_{\alpha_1})$$

$$\alpha_2 \land A(P_{\alpha_2}) \geq \alpha_2 \land B(P_{\alpha_2})$$

$$...$$

$$\alpha_n \land A(P_{\alpha_n}) \geq \alpha_n \land B(P_{\alpha_n})$$

Therefore,

$$\max_\alpha (\alpha \land A(P_{\alpha})) \geq \max_\alpha (\alpha \land B(P_{\alpha})), \text{ and so, } T_A \geq T_B.$$ 

Based on this Lemma, we can prove the main result.

---

Table 4: Truth value computation using proposed method, of LPS associated with example 1

<table>
<thead>
<tr>
<th></th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag 1</td>
<td>0.10</td>
<td>0.30</td>
<td>0.30</td>
<td>0.70</td>
</tr>
<tr>
<td>Bag 2</td>
<td>0.70</td>
<td>0.40</td>
<td>0.30</td>
<td>0.20</td>
</tr>
<tr>
<td>Bag 3</td>
<td>0.70</td>
<td>0.50</td>
<td>0.30</td>
<td>0.30</td>
</tr>
</tbody>
</table>
**B. Theorem 1**

The truth values computed by the proposed method follows the semantic order of the quantifiers. That is, suppose we have three semantically ordered quantifiers $A$, $B$ and $C$ with their membership functions $A(x)$, $B(x)$ and $C(x)$ respectively, each of the form defined in equation (1), and suppose that

$$T(A \text{'s are } P) \geq T(B \text{'s are } P).$$

Then

$$T(B \text{'s are } P) \geq T(C \text{'s are } P)$$

**Proof:**

Let, $T_A = \text{Truth value of the statement, } A \text{'s are } P$, where $A(x)$ is the membership function of the quantifier $A$, along with similar notations for quantifiers $B$ and $C$. Assume that the quantifiers $A$, $B$ and $C$ are semantically ordered. To compute the truth value as shown in equation (8), the quantifiers are transformed to monotonic functions as shown in equation (6). It is straight forward to see that

$$A_1(x) \geq B_1(x) \geq C_1(x), \text{ and} (10)$$

$$C_2(x) \geq B_2(x) \geq A_2(x) \text{ (11)}$$

Now using Lemma 1 and equations (10) and (11),

$$T_{A_1} \geq T_{B_1} \geq T_{C_1} \text{ (12)}$$

$$T_{C_2} \geq T_{B_2} \geq T_{A_2} \text{ (13)}$$

It is given that $T_A \geq T_B$ (14)

We start by computing $T_B$, 


Case 1, $T_A = T_{A_1}$, that is, $T_{A_2} \geq T_{A_1}$

From equation (12) and (13) we have

$T_{A_1} \geq T_{B_1}$ and $T_{B_2} \geq T_{A_2}$. Hence,

$T_{B_2} \geq T_{A_2} \geq T_{A_1} \geq T_{B_1}$, and so

$T_B = \min(T_{B_1}, T_{B_2}) = T_{B_1}$

Case 2, $T_A = T_{A_2}$

Suppose that $T_B = T_{B_2}$.

According to equation (13), $T_{B_2} \geq T_{A_2}$, which would imply that $T_B \geq T_A$. This contradicts the hypothesis of the theorem, restated in equation (14). Hence, if the given condition holds then irrespective of which component defines the truth value $T_A$,

$T_B = \min(T_{B_1}, T_{B_2}) = T_{B_1}$ (15)

Similarly, to compute $T_C$, from equations (12), (13) and (15) respectively we have,

$T_{B_1} \geq T_{C_1}$, $T_{C_2} \geq T_{B_2}$ and $T_{B_2} \geq T_{B_1}$. Thus,

$T_{C_2} \geq T_{B_2} \geq T_{B_1} \geq T_{C_1}$, which implies that

$T_C = \min(T_{C_1}, T_{C_2}) = T_{C_1}$.

Now, since $T_{B_1} \geq T_{C_1}$ and that $T_B = T_{B_1}$ and $T_C = T_{C_1}$, we conclude that,

$T_B \geq T_C$. That is, $T(B\ y's\ are\ P) \geq T(C\ y's\ are\ P)$ and the proof is complete.
From this theorem, it’s clear that the truth values will follow the semantic ordering in all directions along with the quantifiers.

IV. CASE OF EXTENDED PROTOFORMS

Linguistic summaries with simple protoforms: $A$ y’s are $P$ described above have been extended to the protoforms: $A$ y’s are $P$ and $Q$ and $A R$ y’s are $P$ in the past. They can be exemplified by Some of the balls are big and heavy and Some of the big balls are heavy, respectively. As a naming convention, we call summaries of the form $A$ y’s are $P$ and $Q$ as protoform 2 summaries and $A R$ y’s are $P$ as protoform 3 summaries. Both of these extended summaries encapsulate more than one feature of data at the same time, hence containing richer information as compared to simple protoforms.

Along with the truth value, the protoform 3 summaries are accompanied by another metric called the degree of focus [9]. The degree of focus conveys information about how applicable is the summary with respect to $R$ and also enables to discard non-promising summaries. Similar to the case of simple protoforms, we observe a discrepancy when the truth values are computed using Zadeh calculus of linguistically quantified propositions [24] and extend our method to produce more intuitive truth values.

A. Truth value computation using Zadeh calculus of linguistically quantified propositions

Equation (16) and (17) show the method used to compute the truth values of extended protoforms using Zadeh calculus with the formula for degree of focus shown in equation (18)

$$T(A \text{ y's are } P \text{ and } Q) = \frac{1}{N} \sum_{i=1}^{N} (P(y_i) \land Q(y_i))$$ (16)
\[ T(A \ R \ y's \ are \ P) = A \left( \frac{\sum_{i=1}^{N} (P(y_i) \land R(y_i))}{\sum_{i=1}^{n} R(y_i)} \right) \] (17)

\[ d_f(A \ R \ y's \ are \ P) = \frac{1}{N} \sum_{i=1}^{N} R(y_i) \] (18)

where, similar to the simple protforms, \( P(x) \) is the membership function of the summarizer, \( A(x) \) is the membership function of the quantifier and \( N \) the size of the set \( \{y_i | i = 1, 2, ..., N\} \).

For protoform 3 summaries, \( R \) is called the qualifier with its membership function represented by \( R(x) \) and for the case of protoform 2 summaries, \( Q \) is another summarizer along with \( P \). Also, it is worth noting that for protoform 2 summaries, the degree of focus is not applicable since it does not add any information.

In the following, we present some examples to illustrate the discrepancy when the truth values are computed by equation (16) and (17) and show that our method solves the mentioned problem. Table 5 shows the bigness and heaviness of 4 bags of balls containing 10 balls each. We wish to compute truth values of the extended protoform summaries of the form: *A of the balls are big and heavy* and *A of the big balls are heavy*, \( A \) being each of the quantifiers, *Almost None*, *Few*, *Some* and *Many* as defined in Figure 1 in Section II.

From the data presented in the Table 5, intuitively, bag 1 is best described by *Few of the big balls are heavy* and *Few of the balls are big and heavy* while bag 2 by *Almost None of the big balls are heavy* and *Almost none of the balls are big and heavy*. To point out that both of the extended summaries conveys similar but different type of information about the
data, we designed data for bag 3 and 4, such that bag 3 is best represented by *Almost None of the balls are big and heavy* (since there is only one big ball big that is heavy) and *Few of the big balls are heavy* (out of four really big balls, one is heavy). Similarly, for bag 4, the summary *Many of the big balls are heavy* describes it well, as all the four really big balls have higher memberships in heaviness. However, due to the fact only four balls are really big and heavy, the summary *Some of the balls are big and heavy* is equally true but conveys different information.

Table 5: Bigness and Heaviness of 4 bags of balls

<table>
<thead>
<tr>
<th>Bag</th>
<th>Bigness</th>
<th>1.0</th>
<th>1.0</th>
<th>1.0</th>
<th>1.0</th>
<th>1.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heaviness</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Bag 1</td>
<td>Bigness</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Heaviness</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Bag 2</td>
<td>Bigness</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Heaviness</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Bag 3</td>
<td>Bigness</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Heaviness</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Tables 6 and 7 shows the truth values computed using equation (16) and (17), respectively, with the degree of focus shown for protoform 3 summaries in the last column of Table 7.

We observe that for bags 1, 3 and 4, this method computes the highest truth values for both the protoform 2 and protoform 3 summaries that best represents the data. However, in case of bag 2, the truth value of *Few of the big balls are heavy* and *Few of the balls are big and heavy* being highest does not look intuitive, since none of the balls are really heavy (their membership being 0.3). Similarly, for bag 4, even though the truth value of *Some of the balls are big and heavy* is highest, it being very close to *Few of the balls are big and heavy* is non intuitive. This discrepancy arises due to the use of average operator similar to the
case of simple protoforms. We also note, none of the summaries can be rejected on the basis of the degree of focus, since it is fairly high for all of the 4 bags.

**B. Proposed method adapted to the extended protoforms**

We modify the method proposed earlier in this paper to compute truth values of the extended summaries with non-decreasing quantifiers, as shown in equation (19) and (20).

\[
T(A \ y's \ are \ P \ and \ Q) = \max_{\alpha} \left( \alpha \wedge A \left( P_{\alpha}^{Q} \right) \right)
\]

(19)

where,

\[Q_{\alpha} = \{ y_i \in Y \ | \ Q(y_i) \geq \alpha \}, P_{\alpha}^{Q} = \left\{ y_i \in Q_{\alpha} \ | \ P(y_i) \geq \alpha \right\}_{N}\]

\[
T(A \ R \ y's \ are \ P) = \max_{\beta \in [0, \max(R(y_i))]} \beta \wedge \left( \max_{\alpha \in [0,1]} \left( \alpha \wedge A \left( P_{\alpha}^{R_{\beta}} \right) \right) \right)
\]

(20)
\[ d_f^{R_\beta} = \frac{1}{N} \sum_{i=1}^{R_\beta} R(y_i) \text{ such that } y_i \in R_\beta \]  

(21)

where, \( R_\beta = \{ y_i \in Y \mid R(y_i) \geq \beta \} \)

\[ P_\alpha^{R_\beta} = \frac{\left| \{ y_i \in R_\beta \mid P(y_i) \geq \alpha \} \right|}{|R_\beta|}, \text{ for } |R_\beta| > 0. \]

We note that for cases where \(|R_\beta| = 0\), the truth value of the summary for that \( \beta \) cut is set to 0, since there are no objects to be summarized.

To elaborate, for protoform 2 summaries, we modify equation (5) by implementing the ‘and’ between summarizers with a minimum operator. That is, in equation (19), for each object we take the smaller of the memberships in \( P \) and \( Q \), and then compute the truth value of this new set.

For the case of protoform 3 summaries, we take beta cuts of the qualifier data from 0 to the highest membership of the objects in the qualifier \( R \) (i.e., \( \text{max}(R(y_i)) \)). For each of the objects falling in the current beta cut, we follow the procedure for the simple protoforms.

The intuition behind this is that when computing truth values of summaries like *Some of the big balls are heavy*, we should only focus on the balls that are *big* under certain condition, which is the beta cut in equation (20). Next, as shown in equation (21), the degree of focus is computed for the beta cut that produces this truth value, that is, we compute the proportion of objects that have the memberships in the qualifier, \( R \), greater than the value of beta that resulted in the truth value in equation (20). This results in a degree of focus for each summary, unlike the Zadeh calculus method, where there is one degree of focus for
each dataset to be summarized. Also, for summaries with zero truth value, the degree of focus is not applicable since it does not provide any information.

For quantifiers whose membership functions are not monotonic non-decreasing, similar to the simple protoforms, we split the quantifiers as shown in section C and use equation (8) to compute the final truth value. Tables 8 and 9 shows the truth value of the summaries of four bags shown in Table 5, computed using equation (19) and (20) respectively. For bag 2, *Almost none of the big balls are heavy* has the highest truth value with the same degree of focus for all four summaries. In the case of bag 4, the truth value of *Many of the big balls are heavy* is computed to be highest with a degree of focus of 0.36. For this bag, *Some of the big balls are heavy* has a high degree of focus, but it can be rejected on the basis of a very low truth value. It is also worth noting that for bag 3, the truth value of *Many of the big balls are heavy* is computed to be maximum, however, with a very low degree of focus. This is due to the fact that the only really heavy ball has bigness of 0.9. Hence, this summary can be interpreted as *Many balls with bigness above 0.9 are heavy* (which seems to be true).

But we can discard this summary on the grounds of having very small degree of focus. For the same bag, the truth value of *Few of the big balls are heavy* is computed to be 0.8 with a comparatively higher degree of focus of 0.33, which seems to be the correct representation of the balls in bag 3.

For the case of protoform 2 summaries, our method produces expected truth values for all four of the bags. Unlike the method involving the Zadeh calculus, we end up with *Almost None of the balls are big and heavy* as the summary with highest truth value for bag 2 which is in correspondence with the data. Also, for bag 4, *Some of the balls are big and heavy* has a very high truth value in comparison to *Few of the balls are big and heavy* which looks more correct intuitively.
In this work we identified some discrepancies in previous techniques to compute truth values of linguistic protoform summaries of the form Few balls are big, Few of the balls are big and heavy and Few of the big balls are heavy. We solve the problems associated with these methods and show that our solution computes intuitive truth values of linguistic summaries with both simple and extended protoforms. Moreover, we provide a proof to show that our method always produces truth values according to the semantic order of language it represents. Linguistic protoform summaries have been used in numerous applications in the past. A fresh look at them with our new method might provide important observations.

ACKNOWLEDGMENT

This research is supported in part by Center for Eldercare and Rehabilitation Technology at the University of Missouri.
REFERENCES


Chapter 3

Textual Summarization of Events Leading to Health Alerts

Akshay Jain, Student Member, IEEE and James M Keller, Life Fellow, IEEE

Abstract—Extracting information from the sensors installed in the homes of elderly pose a unique set of challenges. Add to it the short amount of time the clinicians and nurses have to analyze this data, and the problem becomes more complicated. A system already in place at an “Aging in Place” facility monitors the activities of residents through multiple non-intrusive sensors and sends alerts on detecting an unusual event. We present an approach to generate textual summaries of events leading to the alerts. We analyze our system using four case studies and also list the comments provided by collaborators in healthcare domain. The system was then iterated to take some of those suggestions into account to give a glimpse of what an ideal system should look like.

I. INTRODUCTION

In the recent times, there has been substantial growth in the development of health monitoring sensors. For instance, monitoring vital signs of patients in hospitals, living routines of elderly whose apartments are equipped with multiple sensors or even that of people just trying to monitor their health with the wearable sensors that have surfaced of late in the market. This leads to the generation of huge amounts of data which is believed to be rich in information content. To make sense of this information rich data, a popular method
is to visualize it graphically and present it to people in the concerned domains, which can be medical personnel in hospitals, nurses etc. But in this method the healthcare professionals are subjected to an added responsibility of understanding the data along with inferring clinical information from it. Along with this, the visualization method pose challenges like getting familiar with it, axis scaling issues and the time to extract information from plots.

A different take to deal with this is to design systems that can identify important patterns in the data and present these in natural language to the concerned audience. This paradigm has been explored in the past by the joint efforts of the Natural Language Generation community and people in health domain. We list a few of them here for reference. The system presented in [1] combines the visualization and the text summarization techniques to efficiently present the medical histories of patients to clinicians. The visual interface presents a snapshot of the patient’s history while the textual summaries of specific events are shown when queried by the user. Their data-to-text system has been evaluated in [2] by comparing the automatically generated textual summaries with raw data (consisting of test results, documents, etc.). In order to assess the usefulness of the system, it was presented to health domain experts, along with the medical histories of the patients. It was concluded that the clinicians were more inclined towards the automatic generated text summaries as compared to the patient history documentation, due to the consistency in the automatic summaries. The system in [3] generates natural language summaries of data in neo natal intensive care units and presents them along with the raw data to the nurses between the shifts. Then, based on a questionnaire posed to the nurses, they conclude that the summaries are understandable, accurate and helpful. Moreover, they show that their system can find “interesting” patterns and summarize them automatically which might be missed by not so experienced professionals. Another example is [4] in which the authors summarize medical histories for patient’s personal use.
The major goal of this work is natural language summarization of sensor data obtained from the homes at TigerPlace, which is an “Aging In Place” facility for elderly at Columbia, MO. The apartments have various sensors installed to monitor the elderly in order to help nurses keep track of their health conditions and assist them if there is a possibility of a bad event. Linguistic Protoform Summaries (LPS) are used to compute the summaries of data which are then modified with natural language rules to appear more intuitive to the end user. A numeric system already in place produces real time alerts (notifications via email) in case a possible unusual activity is detected in any individual sensor data stream (motion sensor, bed sensor etc.). Feedback is then provided by the clinicians on the quality of the alerts. In this work we present four such cases and make a case for how linguistic summaries can help better understand the data. In order get an idea of the preferences of the clinicians, we presented the alert summaries to our collaborators having backgrounds in healthcare. We also list the comments provided by them and modify the summaries accordingly.

II. BACKGROUND

In this section we provide a brief description of the system and the techniques used in this work that have already been developed previously.

A. System at TigerPlace

The apartments at TigerPlace are equipped with motion sensors, bed sensors and Microsoft Kinect cameras to monitor the activities of elderly. In this work we focus on four such apartments which all have multiple motion sensors spanning across the apartment and a bed sensor to monitor restlessness in bed. Each sensor is attached with a numerical alert framework which signals the nurses and researchers about a possible unusual activity in the incoming data stream. An activity is deemed unusual if its distance from the mean of the normal distribution comprising of past two weeks of data is greater than a threshold. The
thresholds are determined experimentally. A more detailed study on the selection of these thresholds can be found in [5]. Also, on receiving the alert email, the clinicians dig into the sensor data and residents medical history, and provide a rating of the alert on a scale of 1 to 5 along with a textual feedback.

**B. Linguistic Protoform Summaries (LPS)**

Linguistic Protoform Summaries are short template based sentences providing a snapshot of data in textual form. For example, a summary focusing on the sizes of balls in a bag might look like, *Most of the balls in the bag are big.* In the sense of Yager [6], they are comprised of a Quantifier (*Most, Few* etc.) conveying information about the quantity of the attribute being summarized, a Summarizer (*big, small* etc.) measuring the feature in concern and a truth value (0 to 1) which signifies the validity of the summary with respect to the data. To generate summaries of the data, we define a set of Quantifiers and Summarizers which are modelled by fuzzy sets used to computed the truth values. Then for each summarizer, the summary with highest truth value is chosen as its best representation. The truth value is computed using the method presented recently in [7]. We now provide a brief description of this method. The truth value of the summaries of the form *A y’s are P* is computed as shown in (1)

\[
T(A \ y's \ are \ P) = \max_{\alpha \in [0,1]} (\alpha \land A(P_\alpha))
\]

where, \(^\land\) is the minimum operator, \(P(x)\) is the membership function of the summarizer \(P\),

\[
P_\alpha = \frac{|\{y_i \in Y \mid P(y_i) \geq \alpha\}|}{N}
\]

is the proportion of objects whose membership in \(P(x)\) is greater than or equal to \(\alpha\) (varies from 0 to 1 in small intervals), \(|.|\) denotes the cardinality of a set and \(A(x)\) is a normal, convex and monotonically non-decreasing membership function of the quantifier \(A\). For quantifiers whose membership function is not monotonically non-
decreasing, it is split into two monotonically non-decreasing functions, $A_1(x)$ and $A_2(x)$ (which is used to compute $\overline{A_2}(x)$) and the truth value is computed as shown in (2). Please refer [7] for more details.

$$T(A y's are P) = T(A_1 y's are P) \land T(\overline{A_2} y's are P)$$  \hspace{1cm} (2)

### III. METHODOLOGY

Figure 1 presents the block diagram of our system.

![System Block diagram](image)

The system already in place generates alerts for individual sensor parameters, such as motion density, bathroom motion, bed restlessness. However, on receiving an alert of one kind, it is useful to look at the other parameters to test the validity of the alerts as well as to have a snapshot of the person’s routine. In fact, we have observed that when the clinicians look at a specific sensor alert, they go back and check the other parameters as well. Taking inspiration from this, we start by considering three parameters, namely, the motion density, the bed restlessness and the bathroom motion.

The motion sensors and bed sensor monitor the activity of the residents inside the apartments round the clock. However, in this study we only focus with activities during the night time, that is 12:00 AM to 06:00 AM. For each night, we measure the motion density, bathroom
motion and bed restlessness. Motion density for each hour is computed by counting the total number of motion sensor hits in that hour divided by the fraction of time spent inside the apartment during that hour. To calculate motion density during the night time, motion density for each hour is accumulated together. The bathroom motion is computed by counting the total number of motion sensor hits in the bathroom during the night time. Bed restlessness is computed by accumulating all the readings of the four sensors installed under the mattress measuring the motion on bed.

The selection of bed restlessness and bathroom motion mentioned above enables to monitor the sleeping condition as well as how many times the person makes a bathroom visit. On the other hand, the motion density tells us whether a person moves around the apartment during the night time. In the following, we explain the different parts of our system to generate textual summaries of the parameters mentioned above which are accompanied with the alerts.

**A. Linguistic Protoform Summaries of the sensor parameters**

For each sensor parameter we generate summaries of the form: *The bed restlessness tonight is a lot higher than most of the nights in the past two weeks*, where, *most* and *a lot higher* is the quantifier and summarizer respectively. We start by defining their membership functions as shown in Figure 2 and 3, respectively.
Corresponding to the alert for which the summary is to be generated, we go back past two weeks and compute the difference between the sensor parameter tonight and each night in the last two weeks. These values are then used to compute the memberships in the summarizers (shown in Figure 3) of each sensor parameter. For each summarizer, we find the quantifier that best represents it, which is the one with highest truth value. In case of two summaries having the same truth value, the summary comprising of the quantifier on the right side in Figure 2 (that is, the one which represents more information) is chosen. The same procedure is carried out for all the three sensor modalities, irrespective of which one produced the alert. Table 1 shows the LPS for the motion density data shown in Figure 4. It is easy to see that the truth value for each summary is in line with the membership functions and data being summarized.

**B. Language Rules**

Intuitiveness is a very important aspect of any Natural Language Generation (NLG) system. The linguistic protoform summaries presented in Table 1 convey the information about the data, however, reading it as it is, might be tiresome for the user. In the NLG domain, this
stage where the information is modified to sound more intuitive is called the document planning, and micro-planning and realization [8]. However, since our summaries are not too complex, we use a simple set of rules to make the presentation of textual summaries intuitive. In the following, we list some of those rules.

In the LPS of Table 1, the summaries with the quantifier *almost none* are useful when comparing two sets of summaries, for instance in [9]. However, when presented to a user, they might be redundant. Hence, for this application, we discard all the summaries with quantifier *almost none*. Another important feature that makes the language intuitive, is the use of appropriate conjunctions. For instance, the use of the conjunction ‘*however*’ when joining two sentences representing information on the opposite end. Similarly, the rules for conjunctions, ‘*Also*’ and ‘*And*’ needs to be defined. Using such rules, the LPS presented in Table 1 produces the following summary: ‘*The bathroom motion tonight is similar to many of the past 14 nights. However, it is lower than a few nights.*’

**IV. CASE STUDIES**

The experimental procedures involving human subjects described in this paper were approved by the Institutional Review Board. We presented four retrospective case studies.
with textual summaries of the form mentioned above to a group of health care professionals comprising of nurses, physicians, etc. to see whether they are helpful in analyzing the alerts. For each case there was a numeric alert, for one of the sensor parameters out of the motion density, bathroom motion and bed restlessness. All of the four alerts were rated very well by the group previously. For each case, the summaries of all three sensor modalities were presented.

The general consensus was that the textual summary of the past events of sensor modalities other than the one which generated the alert is helpful in analyzing the events that lead to it. However, they suggested that for individual summaries, the words representing the quantifier and summarizer membership functions should be more intuitive to a non-technical audience. For example, the use of phrase ‘a lot’ is preferable to ‘significantly’ or instead of saying ‘past 14 days’, calling it ‘past 2 weeks’ would be more helpful. Also, the inclusion of too many summarizers in one summary is not desirable. For instance, in cases where multiple summarizers have a quantifier other than almost none, it might be better to filter or combine some of the summarizers based on the information they are conveying. Moving to the visualization, the group usually analyzes the sensor data using bar plots. However, we presented the textual summaries along with the line plots like the one shown in Fig 4. This led to a discussion on scale of the plot axis which is an important parameter in identifying patterns in the raw data. A single reading at one point can affect the scale of the complete plot. For example, a large scale may prohibit a user to interpret the consistency of sensor parameter. Since presenting the data in form of textual summaries involves a pre-processing step, we believe that they have an added advantage of solving this problem of scale. There were also a lot of suggestions on the inclusion of sensor patterns in the summaries.

Taking cue from the discussion, we modified the “language rules” presented earlier in this section to incorporate the suggestions. Moreover, we found out that many comments entered
by the clinicians described the data trend that lead to the alerts. Therefore, we also developed a simple method to identify the pattern of the data that led to the alert and attached that along with the summary. To detect this pattern, we first median filter the data with a window size of 3 and then move backwards from the last day before the alert until when we find a change in direction. If the number of days with either increasing or decreasing trend are greater than or equal to 3, then we indicate the trend along with the alert summary.

Figure 5 through 8 presents the past 2 weeks data for the bathroom motion, motion density and bed restlessness that lead to the alerts, along with the text summaries for each sensor parameter. Note that the graphs may not accompany the summaries initially in a fielded system. The axis for each line plot is selected in order to have maximum resolution.

A. Case I: Bathroom Motion Alert

The figure below shows the variation of the data that lead to the bathroom motion alert. It is clear to observe that the bathroom motion on the day of alert does not look normal as compared to the previous two weeks. Also the increasing trend in the last few days in bathroom motion and bed restlessness is evident which is also described by the summaries.

B. Case II: Bed Restlessness Alert
In this case also the bathroom motion and bed restlessness on the night of alert are on the higher side than usual and has been increasing for the past some nights. The same has been reported by the automatic text summaries. For the motion density, there are no significant number of days falling under the category of any of the summarizers. This might not be clear from the plot above since the scale is very large for most of the entries due to one very high value on December 13.

C. Case III: Bathroom Motion Alert

In this case, the bathroom motion on the night of alert is higher than about a half and a lot higher than the other half of the nights in past two weeks. Therefore, we combine the two sentences in order to improve readability. However, it doesn’t have any evident trend leading to the night of alert. Hence, we cannot conclude anything to say in the text summary. The motion density in this case is been changing quite a bit, which ends up being summarized as a lot higher and a lot lower than many and few nights respectively. Also, on smoothing the data we observe an increasing trend in motion density in the some of the last nights.

Figure 6: Case 2: The alert was generated by bed restlessness

- The bathroom motion tonight is higher than most of the nights in past 2 weeks with an increasing trend in the past 6 nights.'
- The motion density has been following a decreasing trend in the past 3 nights.'
- The bed restlessness tonight is a lot higher than most of the nights in past 2 weeks with an increasing trend in the past 6 nights.'
days leading up to the alert. The bed restlessness has been decreasing in the past few days, which is reported by the summary.

D. Case IV: Bed Restlessness Alert

In this case the alert was generated due to high bed restlessness which is visible in its plot. However, there is no obvious trend in the past few days, therefore no trend summary has been reported. For motion density, there seems to be an increasing trend for the past three nights, but after filtering the data, this trend disappears, hence the system does not generate any trend summary for motion density. For the case of bathroom motion, the median filtering removes the spikes, after which it is concluded that there is an increasing trend. Also, the

The bathroom motion tonight is similar to many of the nights in past 2 weeks. However, it is lower than a few nights with an increasing trend in the past 6 nights.'

The motion density tonight is a lot higher than almost all of the nights in past 2 weeks. However, it is a lot lower than a few nights with an increasing trend in the past 6 nights.'

The bed restlessness tonight is a lot lower than many of the nights in past 2 weeks with a decreasing trend in the past 3 nights.'

Figure 8: Case 4: The alert was generated by bed restlessness

In this case the alert was generated due to high bed restlessness which is visible in its plot.
bathroom motion has been varying a lot in the past 2 weeks, which can be interpreted by the summary conveying that the bathroom motion is similar to many but lower than a few nights.

V. CONCLUSION

With the help of four case studies we showed that our method has the potential to produce valid and useful textual summaries of the sensor data. Moreover, accompanying the summaries with patterns provides a deeper insight into data and saves the effort of identifying them manually. Another important aspect of such systems is the use of Natural Language that sounds intuitive to the user and can be tailored to clinicians, family members, or the resident. One of the areas of improvements might be to identify more patterns in the data and checking that if a similar pattern led to a health event before. Also, a system which adapts itself to each individual resident and/or sensor is something worth exploring and has the potential of getting over the scale issue faced while visualizing the data graphically.

REFERENCES


Chapter 4

On the use of Linguistic Protoform Summaries as features

4.1 Introduction

In Chapter 3 we saw the use of Linguistic Protoform Summaries (LPS) as a tool to communicate intricate information about the underlying data. However, since the LPS are, at their core, just the macro level representation of data, similarity between two LPS can be considered as calculating similarity between the corresponding datasets. To this end, the authors in [2] introduced a method to compute distance between two LPS and proved that this distance is a metric. This technique is then used to find linguistic prototypes from a set of linguistic summaries [1]. Basically, they group a list of LPS into some specified number of clusters and the linguistic prototypes are the cluster medoids. The distance between the prototypes of two datasets can then be considered as the distance between the data itself. In [3] the authors introduce a method to compute similarity between sets of linguistic summaries.

In this chapter we start by listing some of the methods relevant to our work, in order to compute similarity/dissimilarity between LPS. We then point out some of the shortcomings of these methods which is then followed by potential solutions. We also illustrate the efficiency of our method with the help of synthetic data and then end with some results with real data from the eldercare domain.
4.2 Similarity between LPS using method presented in [2]

In this section we describe the method of [2] in order to compute distance between two LPS. Consider two summaries of the form $Q_1 \ y's \ are \ P_1$ and $Q_2 \ y's \ are \ P_2$ where $Q_1$, $Q_2$ and $P_1$, $P_2$ are the Quantifier and Summarizers respectively, then the similarity between them is given by:

$$sim(Q_1 \ y's \ are \ P_1, Q_2 \ y's \ are \ P_2) = \min(\ sim(P_1, P_2), sim(Q_1, Q_2), sim(T_1, T_2))$$  \hspace{1cm} (1)

Where, $sim(P_1, P_2)$, $sim(Q_1, Q_2)$ and $sim(T_1, T_2)$ are the similarities between summarizers, quantifiers and truth values corresponding to the two summaries, respectively. Note that the method described in [2] deals with distance between summaries even when the summarizers define distinct attributes of the data, for example, $P_1$ may summarize the size of the balls in a bag while $P_2$ describes the weight of the balls. However, in this work we are only concerned with the cases when the summarizers are describing the same attribute.

In the following, we illustrate the computation of different parts of (1). The similarity between the summarizers is given as shown in (2) where $P_1$ and $P_2$ are the membership functions of the summarizers. Equation (2) basically computes the degree of overlap between the two membership functions.

$$sim(P_1, P_2) = \frac{\int(P_1 \cap P_2)}{\int(P_1 \cup P_2)}$$  \hspace{1cm} (2)

Similarly, the similarity between the Quantifiers is computed as shown in (3).

$$sim(Q_1, Q_2) = \frac{\int(Q_1 \cap Q_2)}{\int(Q_1 \cup Q_2)}$$  \hspace{1cm} (3)
Finally, the similarity between the truth values $T_1$ and $T_2$ is given by (4). Note that throughout this chapter the truth values are computed using the method proposed in Chapter 2.

$$sim(T_1, T_2) = 1 - |T_1 - T_2|$$

(4)

where, $| |$ is the absolute value operator.

Computing similarity between summaries using (1) gives rise to situations where it produces the same similarity even when the two summaries have different quantifiers and truth values. We illustrate this with the help of an example: Suppose we have two summaries as shown in Table 1, describing the size of balls in a bag. The membership functions of the summarizers and quantifiers are shown in Figure 1 and 2 respectively.

<table>
<thead>
<tr>
<th>Summary</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some balls are Small</td>
<td>1</td>
</tr>
<tr>
<td>Almost None of the balls are Small</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 1: LPS with truth value

Then the similarity between the two summaries is computed as $\min(1,0,0.6) = 0$, since $sim(\text{Small, Small}) = 1$, $sim(\text{Some, Almost None}) = 0$ (due to no overlap between the two membership functions) and $sim(T_1, T_2) = 1-(1-0.6) = 0.6$. This similarity would be computed the same irrespective of the truth value of the two summaries because there is no overlap between the membership functions of the quantifiers. Also, changing the quantifier will produce same similarity. For example:

$$sim(\text{Some , Almost None}) = sim(\text{Many , Almost None}) = 0$$
However, as per the language semantics, the dissimilarity between (Almost None, Many) should be greater than (Almost None, Some). We provide an alternate method in later sections that attempts to solve this problem.

4.3 Computing similarity between sets of LPS

In reality, it is rarely a case that a single summary can explain a dataset completely. For example, a summary describing the sizes of balls in a bag can be Some of the balls are big, which tells us that the bag has some number of big balls but does not provide the information about the rest of the balls. Hence, to compare two datasets (let say, two bags of balls) using LPS, we need a method to compute similarity between sets of LPS.

We start by describing the method previously presented in [1], then we point out some cases when the results produced by it are not intuitive, which is followed by our technique that has the potential to solve those problems.

4.3.1 Similarity between LPS sets as described in [1]

In [1] the authors present a method to compute similarity between two sets of LPS which is then used to find linguistic prototype sets. To find the sets of summaries
representing the data, first they compute the truth values of all the summaries involving each and every combination of the summarizers and the quantifiers. Then, the summaries with truth value above a threshold are all taken as the best representation of data. Once the set of summaries are computed, the distance between them is calculated as shown in (5), where \( L = \{l_1, l_2..,l_n\} \) and \( M = \{m_1, m_2..,m_n\} \) are the two sets of LPS with similarity between individual summaries computed as shown in (1). Note that the dissimilarity between the two sets is computed as \( 1 – \text{similarity} \).

\[
\text{sim}(L, M) = \frac{\sum_{i=1}^{n} \max_{j=1,...,c} \sim(l_i, m_j)}{n + c} + \frac{\sum_{j=1}^{c} \max_{i=1,...,n} \sim(l_i, m_j)}{n + c}
\]  

(5)

That is, to find the similarity between set \( L \) and \( M \), for each LPS in set \( L \) we find the most similar summary in set \( M \), carry out the same procedure for the summaries in set \( L \), and then take average of all the individual similarity values.

Going back to the question of the selection of individual summaries to form part of LPS set representing a data, we observe that using all the summaries with truth values higher than a threshold may present some discrepancies while trying to understand the data. For example, consider three summarizers - small, medium and big, and four quantifiers – almost none, few, some and many like the ones shown in Figure 1 and 2. Suppose we have a bag containing balls of various sizes and we get the LPS with non-zero truth values as shown in Table 2.

<table>
<thead>
<tr>
<th>Summary</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some of the balls are small</td>
<td>0.6</td>
</tr>
<tr>
<td>Few of the balls are small</td>
<td>0.4</td>
</tr>
<tr>
<td>Few of the balls are medium</td>
<td>0.5</td>
</tr>
<tr>
<td>Almost None of the balls are medium</td>
<td>0.2</td>
</tr>
<tr>
<td>Few balls are big</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 2: Summaries with non-zero truth values describing size of balls in a bag
Now, selecting an optimum truth value threshold can be a tricky task. A high threshold (say, greater than or equal to 0.6) would lead to elimination of all the summaries except *Few balls are big*, while the one on lower side (say, less than or equal to 0.6) would lead to selecting more than one summary for the same summarizers (*small* in case of 0.4), which might not provide any additional information about the data. Also, looking at multiple summaries with same summarizer may confuse the user, as mentioned before in Chapter 3.

Therefore, instead of selecting all the summaries above some threshold to be the representation of data, for each summarizer, we select the summary with the highest truth value. The benefits of this procedure are twofold. First, this avoids the dilemma to manually select the truth value threshold, which can be a non-trivial task, as illustrated before. Second, it is easier to interpret information content of the LPS set when there is only one summary per summarizer. Using this approach the LPS set of Table 2 would appear as shown in Table 3.

<table>
<thead>
<tr>
<th>Summary</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some of the balls are small</td>
<td>0.6</td>
</tr>
<tr>
<td>Few of the balls are medium</td>
<td>0.5</td>
</tr>
<tr>
<td>Few balls are big</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 3: Summaries describing size of balls in a bag with one summary per summarizer

This also makes more sense in terms of traditional feature representation of a dataset in the sense that we can consider each summary as a feature of the data with the number of summarizers as the dimensionality of the feature vector. In the next section we show how this selection of individual summaries modifies equation (5) in order to compute dissimilarity between LPS sets.
4.3.2 Taking Quantifier and Summarizer semantics into account while computing dissimilarity between LPS and sets of LPS

In order to compute similarity between two LPS sets, the method presented in (5) compares each LPS in one set to all of the LPS in the other set. This makes sense while using a truth value threshold to select the individual summaries the LPS set is comprised of. However, if we select the summaries forming part of an LPS set as described towards the end of last section, we have exactly one summary per summarizer in each LPS set. Then, instead of comparing all the summaries to one another in each set, we can only compare the summaries with same summarizer and then aggregate the individual similarities/dissimilarities to find the dissimilarity between two LPS sets. This means that the similarity between summarizers would always be 1 (since they are always same), hence we can remove that term from (1). Now we can write the similarity between the summaries as shown in (6) (Note that we get rid of $P_1$ and $P_2$)

$$sim(Q_1 \text{ y's are P}, Q_2 \text{ y's are P}) = \min( sim(Q_1, Q_2), sim(T_1, T_2))$$

(6)

However, as described in section 4.2, this method would still produce 0 similarity between all the summaries having non-overlapping membership function quantifiers, not depending on the truth values of summary, i.e.,

$$sim(Almost None, Some) = sim(Almost None, Many) = 0$$

which is due to their intersection always being a null set. Hence, this results in similarity being 0. In order to overcome this problem, we define the similarity between quantifiers manually as per their semantic meaning. Table 4 shows heuristically defined similarity between the quantifiers of Figure 2. From Table 4, it is clear that when comparing summaries with same summarizers, the similarity between the ones with quantifiers
Almost None and Some, would be greater than that of with the quantifiers Almost None and Many.

Table 4: Heuristically defined semantic similarity between Quantifiers defined in Figure 2

<table>
<thead>
<tr>
<th></th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Almost None</td>
<td>1</td>
<td>0.66</td>
<td>0.33</td>
<td>0</td>
</tr>
<tr>
<td>Few</td>
<td>0.66</td>
<td>1</td>
<td>0.66</td>
<td>0.33</td>
</tr>
<tr>
<td>Some</td>
<td>0.33</td>
<td>0.66</td>
<td>1</td>
<td>0.66</td>
</tr>
<tr>
<td>Many</td>
<td>0</td>
<td>0.33</td>
<td>0.66</td>
<td>1</td>
</tr>
</tbody>
</table>

Since we now have exactly one summary per summarizer, in order to compute similarity between LPS sets, we modify (5) to only compare summaries with common summarizers. Equation (7) shows this modification where the set similarity is computed to be the average of the similarities between individual summaries, where, \( L \) and \( M \) are the two LPS sets with \( N_S \) summaries each (\( N_S \) being the number of summarizers), \( l_i \) and \( m_i \) represent the individual summaries of LPS sets, \( sim(l_i, m_i) \) is computed using (6) and the similarity between quantifiers is looked up from Table 4.

\[
sim(L, M) = \frac{1}{N_S} \sum_{i=1}^{N_S} sim(l_i, m_i)
\]  

(7)

Some issues still remain with (7) while computing similarity/dissimilarity between LPS sets. In the following we illustrate them with the help of an example, which is followed by a potential solution.

Consider three sets of LPS, summarizing the size of balls in three bags as shown in Table 5. Note that we chose the truth values to be 1 for all the summaries to better illustrate the issue at hand. Intuitively, LPS set 1 seems to be more similar to LPS set 2 than set 3, since the first set consists of all small balls while the third consists of all big
balls and set 2 consists of some big and some medium sized balls. This should be reflected while computing similarity between the three sets.

Equation (7) produces the output in which the similarity between set 1 and set 2 (similarity = 0.22) is smaller than that between set 1 and set 3 (similarity = 0.33). This is due to the fact that while comparing the two sets, the similarity between each pair of individual summaries (corresponding to each summarizer) is assigned equal weights. Therefore, since both sets 1 and 3 have the summary Almost None of the balls are medium, (similarity = 1) this pulls up the overall similarity between the two sets, even though the other two summaries (with summarizers small and big) are less similar. Also, the summary, Almost None of the balls are medium, does not add much information while comparing the two sets since it says what is not present in the two sets. On the other hand, while comparing set 1 and 2, for all three summarizers, even though the quantifiers are closer, each of them have a non-zero dissimilarity, which leads to the two summaries being less far apart. Hence, this problem arises because irrespective of how much information the comparison of two individual summaries are adding while computing dissimilarity between two sets, they are given equal weights in the whole process, which is the classic issue with the mean operator.
We argue that each summary does not impart equal information to an LPS set. For instance, the summary *Many balls are big* conveys more information about the size of balls in a bag than *Almost None of the balls are small*, even though both the summaries are equally true and describing the same bag of balls. To this end, in Table 6, we define weights for each quantifier which are then used while computing dissimilarity between LPS sets. Note that the weights are not constrained to be between [0,1]. We chose this specific set of weights to encode the reasoning that the quantifiers which represent more objects in the data carry more importance. Generally, it is recommended that a set of weights which follows the semantic order of the quantifiers should be used.

<table>
<thead>
<tr>
<th></th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>0</td>
<td>0.3</td>
<td>0.6</td>
<td>1</td>
</tr>
</tbody>
</table>

Now, the similarity between a pair of summaries is weighed according to the maximum of weights of two quantifiers comprising them. Therefore, for each pair, we first compute the similarity between them in the usual fashion using (6), then weigh it with the maximum weight of the two quantifiers.

For example while comparing the LPS sets of Table 5, the weights assigned for the case of set 1 and set 2 would be [1.0, 0.6, 0.6] while that for set 1 and 3 would be [1.0, 0.0, 1.0]. Using these weights, the overall similarity between set 1 and 2 is computed to be 0.18 while that between the set 1 and 3 is 0. This sounds more intuitive since set 2 has some big and some medium balls, which seems closer to set 1 than it does to set 3 which has all big balls. We can also chose the weights to increase equally from the quantifiers having non-zero membership towards the left to the ones at right (For example, 0.25, 0.5, 0.75, 1.0, from *Almost None* to *Many* in this case). This leads to the similarity between set 1 and 2 to be 0.39, while that between set 1 and 3 is computed as 0.11,
which still follows the intuitive order of similarity between the two sets. Therefore, final similarity between LPS sets is not too sensitive to the choice of quantifier weights.

Incorporating this modification, (7) is modified as shown in (8).

\[
sim(L, M) = \frac{\sum_{i=1}^{N_S} \text{sim}(l_i, m_i)W_{q_i}}{\sum_{i=1}^{N_S} W_{q_i}}
\]

where, \( W_{q_i} = \text{maximum of the quantifier weights of summaries } l_i \text{ and } m_i \), all other notations being the same as (7).

Before ending this section, we provide one more modification to (8). Until now we have focused on how assigning weights and semantic similarity between quantifiers helps us to compute intuitive similarities between LPS sets. Likewise, the summarizers face a similar issue. For example: Consider three sets of LPS in Table 7.

<table>
<thead>
<tr>
<th>Summaries</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LPS Set 1</strong></td>
<td></td>
</tr>
<tr>
<td>Many of the balls are small</td>
<td>1</td>
</tr>
<tr>
<td>Almost None of the balls are medium</td>
<td>1</td>
</tr>
<tr>
<td>Almost None of the balls are big</td>
<td>1</td>
</tr>
<tr>
<td><strong>LPS Set 2</strong></td>
<td></td>
</tr>
<tr>
<td>Almost None of the balls are small</td>
<td>1</td>
</tr>
<tr>
<td>Many balls are medium</td>
<td>1</td>
</tr>
<tr>
<td>Almost None balls are big</td>
<td>1</td>
</tr>
<tr>
<td><strong>LPS Set 3</strong></td>
<td></td>
</tr>
<tr>
<td>Almost None of the balls are small</td>
<td>1</td>
</tr>
<tr>
<td>Almost None of the balls are medium</td>
<td>1</td>
</tr>
<tr>
<td>Many of the balls are big</td>
<td>1</td>
</tr>
</tbody>
</table>

Basically, set 1 represents a bag of balls where all of the balls are small, set 2 is for a bag with all the balls medium sized and bag 3 is where all the balls are big. Intuitively, the dissimilarity between set 1 and 2 should be smaller than that between set 1 and 3,
following the semantics of the word *Small, Medium* and *Big*. However, in (8), the summarizers do not play any role in determining the over-all similarity/dissimilarity. Therefore, the sets 1 and 2, and 1 and 3 are determined to be equally similar with similarity equal to 0. We propose a solution to alleviate this problem.

Table 8 presents the semantic dissimilarity between each summarizer of Figure 1, which will then be used to weigh the overall dissimilarity between any two LPS sets.

<table>
<thead>
<tr>
<th>Summarizers defined in Figure 1</th>
<th>Small</th>
<th>Medium</th>
<th>Big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>0.3</td>
<td>0.6</td>
<td>1.0</td>
</tr>
<tr>
<td>Medium</td>
<td>0.6</td>
<td>0.3</td>
<td>0.6</td>
</tr>
<tr>
<td>Big</td>
<td>1.0</td>
<td>0.6</td>
<td>0.3</td>
</tr>
</tbody>
</table>

To incorporate the semantic dissimilarity between summarizers in (8), we use the following procedure. First, for both the sets we find the summaries which have the highest quantifier weight associated with them, then the final dissimilarity between the two sets is weighed according to the dissimilarity between the summarizers of these two summaries (according to Table 8). That is, we weigh the dissimilarity between LPS sets with the dissimilarity between the summarizers describing the highest number of objects in the two sets. For example, for the LPS sets shown in Table 7, while computing the dissimilarity between set 1 and 2, we first calculate the over-all dissimilarity between two sets using (8), which turns out to be 1. Then to find the summarizer weight, we look up the two sets for the summaries with highest quantifiers weights, which are *Many balls are small,* and *Many balls are big* for set 1 and 2 respectively. Then the dissimilarity between the summarizers of these two summaries is computed according to Table 8 (which is equal to 1), which is then used to weigh the final dissimilarity. Using this method produces a dissimilarity of 0.6 between set 1 and 2, and 1.0 between set 1 and 3, which seems more intuitive than both being 1. The reason we deal with dissimilarity in this case is that weighing a similarity of 0 (set 1
and 2, and set 1 and 3 in Table 7) would still produce 0 similarity. We note that this destroys the direct relationship between similarity and dissimilarity. However, we can still compute the overall similarity by 1-dissimilarity which will always be in the [0,1] interval. Also, we do not define the dissimilarity between same summarizers as 0 in Table 8, as it would turn the dissimilarity between all the LPS sets with the same summarizer having highest quantifier weight to 0, which is not always true.

Equation (9) presents the procedure to compute the dissimilarity between two sets incorporating all three modifications described previously in this section (i.e. equation (6), (7) and (8))

\[
dissim(L,M) = \dissim(S_L, S_M) \left( 1 - \frac{\sum_{i=1}^{N_S} \text{sim}(l_i, m_i) W_{qi}}{\sum_{i=1}^{N_S} W_{qi}} \right)
\]  

(9)

where \( L \) and \( M \) are the two LPS sets with \( N_S \) summaries each (\( N_S \) being the number of summarizers), \( l_i \) and \( m_i \) represent the individual summaries of LPS set \( L \) and \( M \) respectively. Here,

\[
sim(Q_1, y's\ are\ P, Q_2, y's\ are\ P) = \min(\ sim(Q_1, Q_2), sim(T_1, T_2))\), with similarity between Quantifiers looked up from the user defined quantifier similarity matrix;

\( W_{qi} = \) maximum of the quantifier weights of summaries \( l_i \) and \( m_i \);

\( S_L, S_M = \) Summarizers of the summary with highest quantifier weight in LPS set \( L \) and \( M \) respectively; and

\( \dissim(S_L, S_M) \) is the dissimilarity between summarizers \( S_L \) and \( S_M \), which is looked up from the user defined summarizer dissimilarity matrix.
4.4 Tests with synthetic data

In this section we present results obtained with synthetic data to illustrate the effectiveness of the modifications described above.

4.4.1 Semantic variation of LPS dissimilarity along with the data

In this test we compare the method presented in this chapter with the method of [1]. To accomplish this we generate synthetic data and see how the dissimilarity between the LPS varies along with the data.

To this end, we start with a bag containing 100 small balls. We then select a number of balls and change their size to be that of medium size balls. We keep on increasing the number of balls to be converted from small to medium until the point when all the balls in the bag are medium. Once the bag consists of all medium sized balls, we increasingly select a number of balls and change their size so that they have the highest membership in summarizer big. Therefore, we start with all the balls in the bag being small, gradually change that to make all the balls medium sized which are then changed until all the balls are big. Along the way, as we change the contents of the bag, we compute the LPS describing the size of balls in the bag using the summarizer and quantifiers whose membership functions are defined by Figure 1 and 2 respectively.

To fix ideas, Figure 3 shows histogram of the size of balls in two bags, with different stages of the experiment shown from top to bottom. The height of each bin represents the number of balls whose size lies in the interval of the bin. The size of all the balls in one of the bags is kept constant throughout the experiment. This is represented by red bars in Figure 3 where the size of balls are distributed between 0 and 0.4. Hence, all the balls in this bag have highest membership in the summarizer Small. The green bars in this figure represents the second bag of balls whose sizes are varied continuously. The
histogram at the top of the figure is at the beginning of experiment when all the balls in the second bag except a very few have highest membership in Small. Therefore, the green histogram is dominant on the left side. As we increase the number of mid-sized balls in this bag, the histogram shifts to the right. The histogram at the bottom is for the last stage when all the balls in the second bag have highest membership in summarizer big. With each histogram plot, we show the LPS set for the second bag (the one with green histogram) as computed by the method described in previous section (i.e. having one summary per summarizer).

We use this synthetic data experiment to compare the two methods to compute dissimilarity between LPS sets. The line plot in red, in Figure 4, is for the method of [1] while the one in blue is using (9). Each point in the plots is the dissimilarity of the LPS for the bag in which all the balls have highest membership in Small to the bag whose contents are continuously varied until the point when all the balls are Big (histogram with red and green bars respectively). Intuitively, the dissimilarity between the LPS sets should increase gradually as we vary the data to contain more and more balls of higher sizes. However, this is not the case for either of the two plots. Nevertheless, unlike the plot for [1] where the variation in the dissimilarity are steep, the dissimilarity computed by the new method varies gradually. The dissimilarity in the new method plot drops a little in the midway around the point 30 when we start replacing some of the medium sized balls with big sized ones.
<table>
<thead>
<tr>
<th>Summaries</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>many of the balls are small</td>
<td>1</td>
</tr>
<tr>
<td>almost none of the balls are mid</td>
<td>1</td>
</tr>
<tr>
<td>almost none of the balls are big</td>
<td>1</td>
</tr>
</tbody>
</table>

LPS for the bag shown in red

---

Figure 3: Illustration showing variation in data used to compare dissimilarity between LPS sets techniques
4.4.2 Computing linguistic prototype of synthetic data

Basically, linguistic prototypes are what cluster medoids are for numerical data, only that the prototypes are now presented linguistically. For example, [1] produces linguistic prototypes of sensor data obtained from the homes of elderly. They use the sensor data collected over several nights recording the sleeping patterns of residents living in an eldercare facility to compute LPS summarizing the data for each night. These LPS for each night are then used to compute linguistic prototypes, which is done by first stacking the LPS of all nights together and then performing single linkage clustering. The cluster medoids, which are also LPS, are called the linguistic prototypes.

We use a method similar to theirs. However, our technique is different in the following ways:

- The dissimilarity between individual LPS and LPS sets are computed using (6) and (9) respectively.
- Instead of stacking all LPS together and then perform clustering, we keep the LPS sets intact and perform clustering over these sets. Therefore, our linguistic prototypes are LPS sets rather than single LPS. All the individual summaries in
each LPS set are part of the same event (for ex: night in the above example), therefore, they are closer to the ‘typical night’ than [1]. We illustrate this with the help of an example, where we compute linguistic prototypes of synthetic data.

Consider 100 bags of balls with each having 100 balls. Out of the 100 bags, let 40 of the bags contain all small balls, 30 contain all medium sized balls and the rest 30 have all big balls. Figure 5 shows graphically the 100 bags of balls. Note that we restrict the sizes of balls in 0 to 1 and the first 40 rows have size on the smaller side, the next 30 are a little big and the last 30 are biggest. We compute the LPS of the form *Most of the balls are small*, with the membership functions of the summarizer and quantifier defined in Figure 1 and 2, respectively. We first compute LPS of each bag and then cluster them using single linkage clustering. Note that the way the data is constructed, for our method, there are always three clusters since we cluster directly on the LPS sets. However, for the method of [1], where the clustering is performed over the list of all the LPS put together, we determine the number of clusters using iVAT images[4].

Table 9 and 10 show the linguistic prototypes produced by the method of [1] and ours (using (6) and (9)), respectively, along with truth values and support. The LPS listed in Table 9 are the medoids of six clusters found in the dataset along with the number of LPS in each one of them is shown as the support. Since each linguistic prototype is just one summary, it is difficult to combine the information presented by each one of them and relate it to the dataset. For instance, the prototypes: *almost none of the balls are big* and *many balls are small* have support equal to 70 and 40 respectively. This conveys that there are 70 bags in which almost no balls are big and 40 for which many balls are small. However, it is not possible to conclude that the bags in which many balls are small have almost no big balls. On the contrary, since the prototypes of Table 10 are
Table 9: Linguistic prototypes of data shown in Figure 5 using the method of [1]

<table>
<thead>
<tr>
<th>Summary</th>
<th>Truth Value</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>almost none of the balls are small</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>almost none of the balls are mid</td>
<td>1</td>
<td>70</td>
</tr>
<tr>
<td>many of the balls are mid</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>many of the balls are small</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>many of the balls are big</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>almost none of the balls are big</td>
<td>1</td>
<td>70</td>
</tr>
</tbody>
</table>

Table 10: Linguistic prototypes of data shown in Figure 5 using our method

<table>
<thead>
<tr>
<th>Summary</th>
<th>Truth Value</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>almost none of the balls are small</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>almost none of the balls are mid</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>many of the balls are big</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>almost none of the balls are big</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>many of the balls are small</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>almost none of the balls are mid</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>almost none of the balls are big</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Data used for comparing the two LPS prototype computation methods. Each row represents a bag of ball with size of each ball color coded.
sets of LPS, it is easy to understand the data and conclude that out of the 100 bags, 40 predominantly have small balls, and 30 have mid-size and 30 big balls. In this section we showed that the linguistic prototypes found out by our method are more intuitive in terms of understanding the data as compared to the method of [1]. The drawback of our method with respect to [1] is that the user is required to specify the similarity/dissimilarity and weights for quantifiers and summarizers. However, we get away with the requirement of specifying a truth value threshold which is used to decide which summaries to keep and discard. The user defined specifications in our method can be considered to be too heuristic, but we require it in order to incorporate the semantics of language into the algorithm. Moreover, since we use absolute quantifiers (having domain [0,1]), they can be easily ported to different applications as it is. Also, since the hard coded inputs can be directly interpreted to the corresponding words, they are not completely arbitrary. We show this in the next section where we demonstrate our method with a real dataset.

4.5 Computing linguistic prototypes of real data

In this test we compute linguistic prototypes of data from eldercare domain. Figure 6 shows in home activity of a person in an aging in place facility monitored over a month through motion sensors installed across the apartment. Each column in the plot represent one day of activity with each hour represented by a square. The colors of the square show the amount of activity while the black patches are the times where a separate algorithm detected that the resident was out of the apartment.

Reference [5] presents a retrospective study of five residents whose activity was being monitored by in home motion sensors. In that paper, the authors presented one month of time period for 3 separate months for all five residents, with a month when the residents were in normal condition, one month for which they were suffering with a
mental illness and then a month after undergoing a treatment for the illness. Figure 7 shows these motion density maps with each row representing a different resident and the columns are the 3 studied time periods. In the paper, with the help of medical records, they concluded that the motion density maps are related with the health condition of the residents.

To this end, we compute linguistic prototypes by the method used in section 4.4.2, and determine whether they convey information similar to what is inferred visually. If we observe the density maps of Figure 7 carefully, we see that most of the difference between the maps is due to out of apartment activity represented by the black patches. Hence, in this work we are only concerned with summarizing the out of apartment activity using LPS. For each day, we compute LPS of the form ‘The person made many short out of apartment visits’ where many and short are the quantifiers and summarizers whose membership functions are shown in Figure 8 and 9 respectively. Note that the quantifiers are defined to be same as the ones in Figure 2.

To be complete, Table 11 and 12 show similarities and the weight of the quantifiers same as defined previously in Table 4 and 6, respectively. Table 13 shows the dissimilarity matrix for the summarizers. In order to compute the linguistic prototypes, we first compute LPS sets describing each day’s out of apartment activity and then
Figure 7: Active density maps of residents with mental illness across three points in time. Vertical axis depicts time of day and the horizontal axis depicts day of the month. Color areas depict time and movement in apartment, black areas depict time out of the apartment.
Almost None  Few  Some  Many

Table 11: Heuristically defined semantic similarity between Quantifiers defined in Figure 9

<table>
<thead>
<tr>
<th></th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Almost None</td>
<td>1</td>
<td>0.66</td>
<td>0.33</td>
<td>0</td>
</tr>
<tr>
<td>Few</td>
<td>0.66</td>
<td>1</td>
<td>0.66</td>
<td>0.33</td>
</tr>
<tr>
<td>Some</td>
<td>0.33</td>
<td>0.66</td>
<td>1</td>
<td>0.66</td>
</tr>
<tr>
<td>Many</td>
<td>0</td>
<td>0.33</td>
<td>0.66</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 12: Heuristically defined semantic Weights of Quantifiers in Figure 9

<table>
<thead>
<tr>
<th>Weight</th>
<th>Almost None</th>
<th>Few</th>
<th>Some</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>0</td>
<td>0.3</td>
<td>0.6</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 13: Heuristically defined dissimilarity between Summarizers defined in Figure 8

<table>
<thead>
<tr>
<th></th>
<th>Short</th>
<th>Mid</th>
<th>Long</th>
<th>Very Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Almost None</td>
<td>0.25</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>Few</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.75</td>
</tr>
<tr>
<td>Some</td>
<td>0.75</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Many</td>
<td>1</td>
<td>0.75</td>
<td>0.5</td>
<td>0.25</td>
</tr>
</tbody>
</table>
cluster them into three clusters using single linkage clustering to find LPS sets representing the ‘typical day’s’ out of apartment activity of the resident. We then modify the LPS using the ‘language rules’ similar to the one described in Chapter 3 to make them sound more intuitive. We describe them briefly in the following.

For each individual summary with quantifier $Q$ and summarizer $P$, the protoform used is of the form: *The resident made* $Q$ $P$ *out of apartment visit*. For example, if the $Q$ and $P$ are *a few* and *short* respectively, then the final summary is *The resident made a few short out of apartment visits*. Now for each LPS set we use the following rules to combine all the individual summaries in short sentences.

- We discard all the summaries with quantifier *almost none*. This is done in order to avoid overwhelming the reader with a lot of information. Also, since the quantifier *almost none* conveys information about the absence of an attribute in the data it might not be always useful to present this to the user.

- Then we sort the individual summaries in the descending order of their quantifier weights. This is done in order to present the summary representing the highest number of objects (out of apartment visits in this case) first. For example, if there are two summaries in the LPS set: *The resident made a few short out of apartment visits*, and *The resident made many long out of apartment visits*, then the summary with quantifier *many* is presented first, followed by the one with *a few*. In case two or more summaries have common quantifier, then the precedence is decided by the summarizers starting from *short* to *very long*.

- The individual summaries are then clubbed together to improve readability. For example, for the two summaries above, the final sentence would be
presented as: *The resident made many long and a few short out of apartment visits.*

- If all the summaries in a set have *almost none* as the quantifier, then this means that there were no out of apartment activities made on that day. Hence, such sets are represented by sentence: *The resident made no out of apartment visit.*

To illustrate the above rules, consider summaries in an LPS set shown in Table 14

<table>
<thead>
<tr>
<th>Q (Quantifier)</th>
<th>P (Summarizer)</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>almost None</td>
<td>short</td>
<td>0.8</td>
</tr>
<tr>
<td>some</td>
<td>mid</td>
<td>0.6</td>
</tr>
<tr>
<td>a few</td>
<td>long</td>
<td>0.9</td>
</tr>
<tr>
<td>many</td>
<td>very long</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Using the rules described above, the LPS set would be converted to the following sentence:

*The resident made many very long, some mid and a few long out of apartment visits.*

Figure 10 through 14 show the density maps for the five residents along with the linguistic prototypes modified with the language rules described above. Note that for each density map, we fix the number of clusters (which are equal to the number of prototypes) to 3. That is, we wish to find three ‘typical days’ during each one month time period. Each prototype has a support associated with it, which says that how many days in the density map are best represented by itself. Also, for all five cases we calculate the dissimilarity between the linguistic prototypes. The method to compute the dissimilarity is described below.
Let $A$ and $B$ denote the set of two linguistic prototypes with the individual prototypes denoted by $a^i$ and $b^j$ respectively for each set. Also, $N_A$ and $N_B$ are the number of prototypes in each set. Then the dissimilarity between $A$ and $B$ is given by (19).

$$
\text{dissim}(A,B) = \sum_{i=1}^{N_A} \frac{\min_{j=1, \ldots, N_B} \text{dissim}(a^i, b^j)}{N_A + N_B} + \sum_{j=1}^{N_B} \frac{\min_{i=1, \ldots, N_A} \text{dissim}(a^i, b^j)}{N_A + N_B}
$$

(10)

Here, $a^i$ and $b^j$ are sets of LPS in prototype sets $A$ and $B$, respectively, and dissimilarity between them is computed using (9). Basically, for each LPS set in the first prototype set ($A$), we find the closest LPS set in second prototype set ($B$). Then the same procedure is carried out for the second prototype set where we find the closest LPS set in first prototype set. The final dissimilarity between the two sets is the average of all dissimilarities.

We use (19) to compute dissimilarities between the linguistic prototypes of density maps shown in Figures 10 to 14. Note that the dissimilarity between the prototypes saying no out of apartment visit and any other prototype is set to 1. While the dissimilarity between two prototypes denoting no out of apartment visit is set to 0 by default. For each of the five cases, column 2 of Table 15 shows the dissimilarities between the linguistic prototypes of the month when the resident was in the normal state (before suffering from mental illness) and the month when they were suffering.

<table>
<thead>
<tr>
<th>Case #</th>
<th>Dissimilarity between before and during mental illness prototypes</th>
<th>Dissimilarity between before and after mental illness prototypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.263</td>
<td>0.200</td>
</tr>
<tr>
<td>2</td>
<td>0.273</td>
<td>0.265</td>
</tr>
<tr>
<td>3</td>
<td>0.071</td>
<td>0.263</td>
</tr>
<tr>
<td>4</td>
<td>0.249</td>
<td>0.246</td>
</tr>
<tr>
<td>5</td>
<td>0.307</td>
<td>0.268</td>
</tr>
</tbody>
</table>
from mental illness, while column 3 shows the dissimilarities between the linguistic prototypes of the normal month and the month after treatment from mental illness.

**Case 1 (Figure 10)**

For this case we can see that visually the density maps for the month before and after mental illness look very different from the month when the person was suffering from illness. This can also be observed from the linguistic prototypes shown beneath the density maps. The prototypes for both before and after mental illness time periods describe that the person makes a lot of out of apartment visits with varying duration. However, while suffering from mental illness he/she tends to stay inside the apartment which is denoted in the linguistic prototype with no out of apartment visit made for 18 days (shown as support). Moreover, the dissimilarity between the linguistic prototypes as computed by (19) (row 2 in Table 15) also conveys similar information where the dissimilarity between before and during is larger than that between before and after mental illness.

![Case #1: June 29 – July 29, 2010](image1.png)  
![Sept 4 – Oct 3, 2010](image2.png)  
![Jan. 10 – Feb. 11, 2011](image3.png)

<table>
<thead>
<tr>
<th>Before mental illness</th>
<th>The resident made a few short, a few mid, a few long and a few very long out of apartment visits</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The resident made some mid and a few long out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>The resident made some mid and some long out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td>During mental illness</td>
<td>The resident made no out of apartment visits</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>The resident made a few short and a few mid out of apartment visits</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>The resident made a few short and a few long out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td>After mental illness</td>
<td>The resident made a few short, a few mid, a few long and a few very long out of apartment visits</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>The resident made no out of apartment visits</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>The resident made a few very long out of apartment visits</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 10: Density maps for resident case # 1 with the linguistic prototypes for the month before, during and after mental illness
Case 2 (Figure 11)

In this case, unlike case 2, the person happens to make more out of apartment visits during mental illness as compared to before and after treatment. Even though this is not very evident from the linguistic summaries, they still convey somewhat similar information where there are more instances of out of apartment activity in the prototypes of the month during mental illness. The dissimilarity between the prototypes of the month before and during is only slightly higher than that between before and after mental illness. We suspect that the reason behind it is the averaging of all the individual dissimilarities.

Case 3 (Figure 12)

This is an interesting case, where like the visual maps, the linguistic prototype suggest that the month before and after mental illness are similar to each other, but both are very different from the month during mental illness. However, the dissimilarity between...
the month before and during mental illness (0.07) is much smaller than that between before and after (0.26). We suspect that the reason behind this is twofold. First, the method of \((19)\) finds the closest prototypes between the two prototype sets. Since, both before and during mental illness have no out of apartment visit prototype (dissimilarity between two no out of apartment activity prototypes is 0), the dissimilarity for the whole prototype set is pulled down to small value due to average operator. Similarly, the month after mental illness does not have any no out of apartment activity prototype, which pulls up the dissimilarity including this month to a high value (dissimilarity between a no out of apartment activity prototype and any other prototype is 1 by default). Second, since \((19)\) does not take the support of the prototypes in account while computing dissimilarity between them, the fact that the prototypes of the month before mental illness has only 1 no out of apartment activity prototype while the one during mental illness has 14 such days is neglected, which does
not seem correct intuitively and might be the reason for the dissimilarity between prototype sets to be so low.

**Case 4 (Figure 13)**

In this case the linguistic prototype convey information similar to what is observed visually. That is, the prototypes of the month after mental illness describe more out of apartment visits than the month during mental illness, but lesser than the month before. Also, the dissimilarities for the months before and during mental illness is very similar to that between before and after. This might be an artifact of the method, or it may due to the fact that the month during mental illness does not appear very different from the other two.

![Figure 12: Density maps for resident case # 3 with the linguistic prototypes for the month before, during and after mental illness](image)

![Figure 13: Density maps for resident case # 4 with the linguistic prototypes for the month before, during and after mental illness](image)
Case 5 (Figure 14)

The linguistic prototypes of this case very clearly suggest that the month during mental illness the resident did not spend much time outside the apartment. Even after the treatment, the out of apartment activity did not return to the normal state which is also suggested by the linguistic prototypes. Table 15 also convey similar information where the dissimilarity between the linguistic prototypes of the month before mental illness and during is higher than that between before and after.

<table>
<thead>
<tr>
<th>Before mental illness</th>
<th>The resident made many short, a few long and a few very long out of apartment visits</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The resident made many short and a few long out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>The resident made some long and a few short out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td>During mental illness</td>
<td>The resident made a few short out of apartment visits</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>The resident made no out of apartment visits</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>The resident made a few short and a few mid out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td>After mental illness</td>
<td>The resident made a few short out of apartment visits</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>The resident made some short and some mid out of apartment visits</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>The resident made no out of apartment visits</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 14: Density maps for resident case # 5 with the linguistic prototypes for the month before, during and after mental illness

From the five case studies shown above, we illustrated that the linguistic prototypes convey information similar to what is observed visually. Also, even though the dissimilarity calculation method between prototypes needs some crucial improvements,
it produces sensible outputs. In order to adapt our method of computing linguistic prototypes to this real data, we only needed to define the dissimilarity matrix for the chosen summarizers. Therefore, it’s reasonable to say that even though our method requires some hard coded specifications, modifying it for other applications does not need a lot of effort.

References


Chapter 5

Conclusions and Future Work

In this work we presented three important aspects of Linguistic Protoform Summaries, namely, computation of truth values accompanying the summaries, the application of LPS to a real world setting such as eldercare and their use as features of data. We see tremendous potential in Linguistic Protoform Summaries and generally in data to text systems.

In Chapter 2, we proposed a new method to compute truth values of both simple and extended LPS. We proved the robustness of our method with the help of a mathematical proof. Then in Chapter 3, we illustrated the usefulness of LPS in designing a data to text system, keeping the requirements of concerned audience in mind. We showed the potential of LPS through four case studies. This system may use improvements in various aspects such as the structure of protoforms and the type of patterns being recognized and summarized in text. Another area of improvement could be to identify patterns and relate them to events in the past in order to help healthcare professionals decide on providing future care.
At last in Chapter 4, we presented a case for the use of linguistic protoform summaries as features of data. LPS features have an advantage over their traditional numeric counterparts in the sense that they are more intuitive to understand. However, since LPS has language semantics associated with them, commenting about their relative dissimilarity is not a trivial task. We illustrated this by pointing out at some shortcomings in the current methods and presenting a possible solution. However, the proposed technique does not maintain the metric property of the previous method. Since the paradigm to use LPS as features is still at an early stage, we developed a methodology using synthetic data to better understand it. The potential of this method for time series comparison is then showed with a real data example. Certainly, there is a lot of room for improvement. For instance, a technique to assign the semantic similarity/dissimilarity between summarizers and quantifiers automatically would be something to be looked into. Also, the method to compute dissimilarity between LPS sets can be improved to produce more intuitive output without too much user defined parameters. The clusters in Figures 10 to 14 of Chapter 3 seems to be locally biased. That is, one cluster out of three is very dense while the others have very less support. Some work could be done to find possible reasons and solutions to this. A method that takes the support into account while computing dissimilarities between prototype sets is highly desirable. Lastly, throughout this work, the membership functions are designed manually. A method to learn these autonomously from a training set would be highly beneficial. Such a method could then be extended in order to tailor the membership functions to each application/user separately.