Information Systems 67 (2017) 114–125

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Linguistic summarization of event logs – A practical approach

Remco Dijkman*, Anna Wilbik

School of Industrial Engineering, Eindhoven University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 22 July 2016
Revised 28 March 2017
Accepted 31 March 2017
Available online 4 April 2017

Keywords:
Linguistic summarization
Business process
Event log
Data mining
Similarity clustering

ABSTRACT

The amount of data that is generated during the execution of a business process is growing. As a consequence it is increasingly hard to extract useful information from the large amount of data that is produced. Linguistic summarization helps to point business analysts in the direction of useful information, by verbalizing interesting patterns that exist in the data. In previous work we showed how linguistic summarization can be used to automatically generate diagnostic statements about event logs, such as ‘for most cases that contained the sequence ABC, the throughput time was long’. However, we also showed that our technique produced too many of these statements to be useful in a practical setting. Therefore this paper presents a novel technique for linguistic summarization of event logs, which generates linguistic summaries that are concise enough to be used in a practical setting, while at the same time enriching the summaries that are produced by also enabling conjunctive statements. The improved technique is based on pruning and clustering of linguistic summaries. We show that it can be used to reduce the number of summary statements 80–100% compared to previous work. In a survey among 51 practitioners, we found that practitioners consider linguistic summarization useful and easy to use and intend to use it if it were commercially available.

1. Introduction

More and more data is being produced during the execution of business processes. This data potentially provides a valuable source of information that can be used to improve the performance of business processes. Among other things, information can be extracted from the data to help diagnose and resolve problems that may exist, such as bottlenecks, particular types of customer cases that are especially hard or expensive to handle, and mistakes that are made at specific points in the process. However, due to the large amount of data that is available, finding relevant information and patterns in data is a chore that is infeasible without proper automated support.

Linguistic summarization is a technique that can help people to extract information from data [1–3]. In particular, linguistic summarization can be used to find pre-defined patterns in data and verbalize those patterns. The verbalized patterns are of the form: ‘Q cases [for which condition C held] had property P’, where Q is a quantifier, such as ‘a few’ or ‘many’, and the part between the square brackets is optional. Quantifier Q and linguistic values C and P are modeled with fuzzy sets [4]. Thus linguistic summarization can be used to automatically produce statements such as ‘many cases that contained a sequence like <receive request, forward request> had a long throughput time’ and ‘most cases that concerned a building application had the property that they were rejected’. These statements have great value for business analysts, who can use them to quickly zoom in on operational problems that exist in a business process.

In previous work we presented a technique for linguistic summarization of business process event logs [5,6]. However, this technique had as a drawback that it produced hundreds to thousands of summary statements for a single log. This number was so large that it was not useful for human interpreters. One of the issues that causes this problem, is that there may exist many small variations of the execution of a case. Each of those small variations potentially leads to a separate statement, thus generating the impractically large summaries.

Therefore, this paper proposes a technique that generates far more concise summaries that are useful in practice. We evaluate the technique by both determining to which extent it can reduce the number of summary statements and by determining whether practitioners would use the technique.

The technique contributes to previous work [5,6], by:

- enabling the creation of summary statements that contain conjunctive conditions;

* Corresponding author.
E-mail addresses: r.m.dijkman@tue.nl (R. Dijkman), a.m.wilbik@tue.nl (A. Wilbik).

http://dx.doi.org/10.1016/j.is.2017.03.009
0306-4379/© 2017 Elsevier Ltd. All rights reserved.

Table 1
Example business process log.

c_id	e_id	e_name	c_customer	e_start	e_end	e_resource
1	11	Register	Mr. Smith	9:00	9:10	John
1	12	Revise	Mr. Smith	9:20	9:25	Susan
2	21	Register	Ms. Smith	9:05	9:17	John
2	22	Accept	Ms. Smith	9:45	10:15	Susan
2	23	Archive	Ms. Smith	10:20	10:30	John
3	31	Register	Mr. Johnson	11:00	11:08	John
3	32	Reject	Mr. Johnson	11:20	11:45	Susan
3	33	Archive	Mr. Johnson	11:45	11:55	John

- exploring different measures of similarity to determine which summary statements are similar enough to be clustered and presented as one summary statement; and
- developing an efficient algorithm for exploring and pruning the large space of possible summary statements.

Consequently, the contribution of this paper is primarily found in Section 4.2, 4.2, 5, and 6, which present the algorithms for clustering and pruning, their evaluation in combination with different similarity metrics, and their practical evaluation.

Against this background the remainder of this paper is structured as follows. Section 2 presents preliminary definitions required in the paper. Section 3 presents our technique for linguistic summarization of business process event logs, which has also been covered in previous work. Section 4 introduces the efficient algorithm for linguistic summarization, which explains the way in which the space of possible summary statements is built, clustered and pruned. Section 5 evaluates the algorithm on an event log from practice. Section 6 evaluates the usefulness of linguistic summarization of event logs in practice. Section 7 presents related work and Section 8 the conclusions.

2. Preliminaries

To define an event log, we require the definition of a sequence.

Definition 1 (Sequence, Length, Containment)**.** Let Σ be a non-empty finite set of elements. Then a sequence σ of length n over elements from Σ is a mapping $\sigma : \{1, 2, \ldots, n\} \rightarrow \Sigma$. Note that $|\sigma| = n$. We also represent $\sigma(i)$ as σ_i and σ as $\sigma_1\sigma_2 \ldots \sigma_n$. Σ^* is the set of all sequences over Σ.

To count the number of times σ contains υ, we define the function $\text{contains}(\sigma, \upsilon) = |\{i | i \in \{1, \ldots, |\sigma| - |\upsilon|\} \wedge \sigma_i \ldots \sigma_{i+|\upsilon|} = \upsilon\}|$

A business process log is a collection of cases, each of which is associated with a sequence. In this paper we use the notation introduced by van der Aalst [7]. Detailed definitions can be found in his book. In these definitions we use the more specific set of elements $\mathcal{E}$, which represents the set of possible business events, to generate sequences over.

Definition 2 (Case, Event Sequence, Log, Attribute)**.** Let $\mathcal{C}$ be the set of all cases and $\mathcal{E}$ be the set of all events. The event sequence of a case $c \in \mathcal{C}$, denoted $\hat{c}$, is defined as $\hat{c} \in \mathcal{E}^*$. A log L is a set of cases, $L \subseteq \mathcal{C}$. Let $\mathcal{A}$ be a set of attributes and $\mathcal{V}$ be a set of attribute values, where $\mathcal{V}_a$ is the set of possible values of the attribute $a \in \mathcal{A}$. For some attribute $a \in \mathcal{A}$ and case $c \in \mathcal{C}$, $\#_a c \in \mathcal{V}_a$ is the value of attribute a in case c, $\#_a e \in \mathcal{V}_a$ is the value of attribute a in event e.

For illustrative purposes, Table 1 shows a log represented as a table. The log contains the cases $\mathcal{C} = \{1, 2, 3\}$, the events $\mathcal{E} = \{11, 12, 21, 22, 23, 31, 32, 33\}$, and the attributes $\mathcal{A} = \{\text{c\_id}, \text{e\_id}, \text{e\_name}, \text{c\_customer}, \text{e\_start}, \text{e\_end}, \text{e\_resource}\}$, where we prefixed attributes of cases with 'c' and attributes of events with 'e'. The table shows the values of the attributes. For example, $\#_{\text{c\_customer}}1 = \text{Mr.Smith}$ and $\#_{\text{e\_start}}11 = 9:00$.

Definition 3 (Fuzzy set, Intersection, Union)**.** Let X be a universe of discourse. Then the fuzzy set S in X is defined as a set of pairs $(x, \mu_S(x))$, where $\mu_S : X \longrightarrow [0, 1]$ is the membership function of S and $\mu_S(x) \in [0, 1]$ is the grade of membership (or the truth value) of an element $x \in X$ in the fuzzy set S.

Let A and B be two fuzzy sets in X. The intersection of A and B $(A \cap B)$ is $\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x)$, for each $x \in X$ where "$\wedge$" is the minimum operation.

Let A and B be two fuzzy sets in X. The union of A and B $(A \cup B)$ is $\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x)$, for each $x \in X$ where "$\vee$" is the maximum operation.

For example, we can have a fuzzy set of hot temperatures over the universe of discourse of temperatures $X = \{-50, \ldots, 50\}$ (in degrees centigrade) with $\mu_{hot} = 0$, if $x < 25$; $\mu_{hot} = (x - 25)/5$, if $x \geq 25 \wedge x < 30$; $\mu_{hot} = 1$, if $x \geq 30$. For this fuzzy set a temperature of 29 has a truth value of 0.8 for being hot. While we use the minimum and maximum operation to compute the membership of intersections and unions in this paper, a different t-norm or t-conorm can be used as well [8].

3. Linguistic summaries of event logs

A linguistic summary is a textual representation of patterns that may exist in a business process execution log. A linguistic summary consists of statements that are created according to predefined templates, which are also called protoforms. We focus on the approach that was proposed by Yager [9] and then improved and implemented by Kacprzyk et al. [10]. In previous work we adapted this approach to create summaries for business process event logs [5]. Accordingly, we distinguish the following protoforms:

The **simple protoform**, which is expressed as:

Q cases had the property P

The **extended protoform**, which is expressed as:

Q cases that met the condition R had the property P

In these protoforms:

- a **quantifier** Q is a linguistic value that describes quantity [11]. We focus here on so-called proportional quantifiers or relative quantifiers, such as *many, most*, and *almost all*.
- a **summarizer** P is a linguistic value for an attribute $a \in \mathcal{A}$. Using a summarizer, we could, for example, speak of *short throughput time* or *high cost*. We consider the sequence of the activities that were performed for a case as a special type of attribute, such that summarizers of the form 'the case contained the sequence σ' can also be used.
- a **qualifier** R is also a linguistic value for an attribute $a \in \mathcal{A}$, similar to a summarizer P, but the role of a qualifier is to define a subset of the log L and narrow the scope of the summary. Qualifiers of the form 'the case contained the sequence σ' can also be used.

Here, a linguistic value is a natural language term, e.g. "young" for the linguistic variable "age". A linguistic value is associated with a membership function, usually a fuzzy set, that defines the correspondence of this linguistic value with numerical values.

Using these different protoforms, we can make various linguistic statements about event logs. An example of a statement that follows the simple protoform is: '*most* cases had *a short throughput time*'. In this statement, the quantifier *most* applies to cases that have the property *short* for the attribute *throughput time*. An