Advanced Engineering Informatics 33 (2017) 388–396

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Extracting failure time data from industrial maintenance records using text mining

Kazi Arif-Uz-Zaman *, Michael E. Cholette, Lin Ma, Azharul Karim

Science and Engineering Faculty, Queensland University of Technology (QUT), 2 George Street, Brisbane, QLD 4000, Australia

ARTICLE INFO

Article history:
Received 31 May 2016
Received in revised form 10 November 2016
Accepted 21 November 2016
Available online 9 December 2016

Keywords:
Text mining
Work orders analysis
Naïve Bayes
Support vector machine

ABSTRACT

Reliability modelling requires accurate failure time of an asset. In real industrial cases, such data are often buried in different historical databases which were set up for purposes other than reliability modelling. In particular, two data sets are commonly available: work orders (WOs), which detail maintenance activities on the asset, and downtime data (DD), which details when the asset was taken offline. Each is incomplete from a failure perspective, where one wishes to know whether each downtime event was due to failure or scheduled activities.

In this paper, a text mining approach is proposed to extract accurate failure time data from WOs and DD. A keyword dictionary is constructed using WO text descriptions and classifiers are constructed and applied to attribute each of the DD events to one of two classes: failure or nonfailure. The proposed method thus identifies downtime events whose descriptions are consistent with urgent unplanned WOs. The applicability of the methodology is demonstrated on maintenance data sets from an Australian electricity and sugar processing companies. Analysis of the text of the identified failure events seems to confirm the accurate identification of failures in DD. The results are expected to be immediately useful in improving the estimation of failure times (and thus the reliability models) for real-world assets.

1. Introduction

Organisations typically record a large amount of data regarding their assets to support accounting and basic analysis of maintenance costs. Most manufacturing plants and maintenance departments use computerized maintenance management systems (CMMS) to keep records of all maintenance activities performed on the asset [1]. However, many organizations have correctly noted that a simple analysis of maintenance activities and costs is not sufficient to determine the optimal maintenance policy. Therefore, there is a desire to exploit the vast quantities of historical asset data to develop more sophisticated analytics, including asset reliability models. Yet, asset data in CMMSs are often collected in an approach that is inconsistent with reliability modelling, focusing on maintenance record keeping and accounting and not with the specific intent of identifying asset failure times [2].

To see why data collection practices are often insufficient to identify failure times, consider a typical example of a definition of asset failure: *the inability of an asset to perform the required function at a given time*. One interpretation of this which may be applied to historical data is that a "failure" is an unplanned event where the organization is forced to maintain the asset. Often, organizations maintain record on various types of work performed on the asset, called *work orders* (WO, in some cases, *maintenance logs*), and the times when the asset was not functioning, often referred to as *downtime data* (DD). Both of these datasets only possess part of the information needed to define a failure event. WO tags typically state if the work is a result of a "defect" or if it is "unplanned" but do not indicate if the asset was "down" as a result (i.e. a "failure"). On the other hand, DD contains asset stoppage time, i.e. "when", but the information as to *why* it was down is often contained in the difficult-to-analyse free text descriptions. Such free texts often detail the work carried out in the downtime without clearly indicating if the work was unplanned or a part of the routine maintenance. Thus, each dataset is incomplete from an analysis point of view, where one needs to know both when the asset is down and if this downtime is unplanned.

In order to link these two datasets, some organizations may use different data tags or identification numbers to link downtime events and the work orders completed during those events. However, in many historical databases (especially for long-lived assets) such a detailed linking is unreliable or not present. One might also

* Corresponding author.
E-mail address: arif.kaziarifuzzaman@hdr.qut.edu.au (K. Arif-Uz-Zaman).

http://dx.doi.org/10.1016/j.aei.2016.11.004
1474-0346/

attempt to link WO or DD data using dates. However, the WO dates are often unreliable and one must make assumptions about which WO was completed during each DD and/or which WO(s) forced the downtime. A possible remedy to the unreliability of the date linking is analysing the free text descriptions in both the WOs and the DD, since these texts are typically consistently completed in both WO and DD. This suggests that text analytics techniques may prove useful in extracting more accurate failure time information than either DD data or WOs alone.

There are a few studies that analyse text in industry maintenance logs. Devaney et al. [3] proposed analysing the free texts from maintenance logs using a domain ontology. While the authors proposed an analysis framework based on the construction of a case library, no case study was presented on real-world data. Edwards et al. [4] categorised maintenance logs using clustering algorithm on a small data subset and labelled the data manually as failure or nonfailure. Sipos et al. [5] used component replacement data (and assume that each replacement constitutes a failure) and operational logs to construct a classifier that can anticipate the imminent failure of the equipment. Trappey et al. [6] developed a back-propagation artificial neural network (BP-ANN) based prediction model to detect transformer's potential failure. Model performance was tested using various operating condition data from Australia and Taiwan power companies. Moreover, Moreira and Junior [7] proposed a method of performing prognostics on aircraft component based on SVM classification algorithm. Flight data and maintenance logs have been used to classify the training data into healthy and unhealthy states. The degradation index is finally created from the classification result to prepare future schedule of aircraft maintenance.

In this paper, a methodology based on text classification techniques [8,9] is proposed to identify accurate failure times from WO and DD. Since urgent maintenance can be clearly identified in WOs, the WO text descriptions are used to construct a classifier that will be used to separate "failure" and "nonfailure" DD descriptions. The developed method is subsequently applied to two case studies: one in power generation and one in the Australian sugar industry. To the best of the authors' knowledge, this paper represents the first application of text mining to fuse multiple maintenance data sources to more accurately identify historical failure times.

The remainder of this paper is organized as follows: Section 2 details the proposed information extraction methodology; Section 3 discusses the experiments and results obtained in the case study; and Section 4 shows the conclusions and future directions.

2. Methodology

The overall methodology proposed in this paper can be seen in Fig. 1. First, the WO and DD texts are pre-processed and the WOs are labelled as *urgent and unplanned* (i.e. potentially failure) or *nonfailure*. The WO and DD text fields are cleaned and a keyword dictionary is developed from the WOs. These labels and the keyword dictionary are used to construct a classifier that separates free text associated with failures from those that are associated with nonfailure maintenance. The constructed classifier is subsequently applied to the free text of the DD to label each downtime event as a failure or nonfailure. The details of the each step will be discussed in the following subsections.

2.1. Work order labelling and data preprocessing

There are two methods that one can use to label the WOs: manual and automatic. The manual approach consists of soliciting expert opinion to decide if each WO is a potential failure or not. The automatic approach consists of using information contained in the work order to decide if it is (or is not) unplanned directly, without the assistance of an expert. Typically, WO data sets contain tags that indicate the urgency and source of the maintenance request. Thus, these tags can be used to directly label each WO as planned/unplanned and urgent/non-urgent. Based on our working definition of failure, if the WO is unplanned (e.g. result of a "defect") and urgent (i.e. "high priority") the work order is considered to describe a potential failure event. Thus, the free texts will likely use words that the organization would use to describe a failure.

After this labelling, the free texts used in both WO & DD are preprocessed to construct a keyword dictionary. Usually, maintenance data contain a large proportion of non-informative text that needs to be cleaned by removing unwanted space, numbers, punctuation and non-discriminating words (i.e. *stop words*). A series of typical text cleaning process has been used to clean free text [10–12]. At the beginning of this process, all the free texts are transformed into lower case followed by removing numbers, punctuation, stemming and extra spaces between the keywords. Filler words such as "to", "and", etc., are removed from the data sets. In addition some keywords that are common but non-discriminating are eliminated (e.g. "system" or "unit").

While there are a variety of methods for generating text features (e.g. [13–15]), the raw text is transformed into a *bag-of-words* representation in this study. This representation ignores the order in which the terms appear and provides only a variable indicating whether the term appears at all. It is then necessary to transform the terms into a matrix form that machine learning algorithms can understand. This can be done by splitting the cleaned text documents into individual words, which is called *tokenization*. The classifier requires data in the form of matrix where each row contains a document and each column presents a keyword (Here keywords are the all the tokens/terms are stored within the dictionary).

The end result of this process is a (sparse) document term matrix arranged into a data structure that can be used to train the classifier.

2.2. Text classification

Text classification is the task of automatically sorting a set of documents into categories from a predefined set and has been applied in many settings including, email filtering [8,13], topic categorization [8], as well as document indexing and clustering [16]. Several researchers have constructed text classifiers based on case-based reasoning (CBR) systems [3,17,18], clustering algorithms & pattern discovery [4,19,20] and Bayesian approaches [21]. These approaches mainly focused on identifying relevant features regarding the correlation of words occurring in documents as well as comparing text classifier performances with traditional ones.

A wide variety of classifiers have also been applied to text mining, e.g., Naïve Bayes (NB) [8,14], Support Vector Machine (SVM) [7,9,22–24], k-nearest neighbour (kNN) [13], decision trees [25], centroid-based classifiers [13], artificial neural network (ANN) [22]. Joachim [23] found that SVM scales well, has good performance on large data sets and outperforms NB and kNN substantially. In similar research, Basu et al. [22] compared SVM and ANN and their result showed the superior performance of SVM on reduced feature set.

Following this literature, two popular classifiers will be utilized in this work: Naïve Bayes (NB) and Support Vector Machine (SVMs) classifiers. In the following, each classifier is briefly described.

2.2.1. Naïve Bayes

A Naïve Bayes (NB) classifier uses an estimation of the joint probabilities of words by using Bayes' law and assumes that the classes are independent of each other (Naïve assumption). The