MaintNet: A Collaborative Open-Source Library for Predictive Maintenance Language Resources

Conference Paper · January 2020

DOI: 10.18653/v1/2020.coling-demos.2

Farhad Akhbardeh

13 PUBLICATIONS **14** CITATIONS

SEE PROFILE

Travis J. Desell

Rochester Institute of Technology

121 PUBLICATIONS **966** CITATIONS

SEE PROFILE

MaintNet: A Collaborative Open-Source Library for Predictive Maintenance Language Resources

Farhad Akhbardeh, Travis Desell, Marcos Zampieri
Rochester Institute of Technology, United States
{fa3019, tjdvse, mazgla}@rit.edu

Abstract

Maintenance record logbooks are an emerging text type in NLP. An important part of them typically consist of free text with many domain specific technical terms, abbreviations, and non-standard spelling and grammar. This poses difficulties for NLP pipelines trained on standard corpora. Analyzing and annotating such documents is of particular importance in the development of predictive maintenance systems, which aim to improve operational efficiency, reduce costs, prevent accidents, and save lives. In order to facilitate and encourage research in this area, we have developed MaintNet, a collaborative open-source library of technical and domain-specific language resources. MaintNet provides novel logbook data from the aviation, automotive, and facility maintenance domains along with tools to aid in their (pre-)processing and clustering. Furthermore, it provides a way to encourage discussion on and sharing of new datasets and tools for logbook data analysis.

1 Introduction

With the rapid development of information technologies, engineering systems are generating increasing amounts of data that are used by various industries to improve their products. Maintenance records are one such type of data. They typically consist of event logbooks which are collected in many domains such as aviation, transportation, and healthcare (Tanguy et al., 2016; Altuncu et al., 2018). The analysis of maintenance records is particularly important in the development of predictive maintenance systems, which can be used to prevent accidents and reduce maintenance costs (Jarry et al., 2018).

Maintenance record datasets generally contain free text fields describing issues (or problems) written in non-standard language with many abbreviations and domain specific terms, as in the instances presented in Table 1.

ID	Job Code	Report Date	Problem
211052	7130	8/28/2012	DOING MX PERFORM T/O @ 3200, MP WON'T GO ANY HIGHER THAN 24
221313	7200	4/7/2015	FRONT R/H BAFFLE WORN THROUGH FROM MUFFLER SHROUD NEED NEW BFFLE.
211585	550	4/10/2015	LACING CORD LOOSE ON SCAT TUBING + IGN LEAD TO FRAME, R/H SI, NEED @ ENG #2.
221958	7250	4/11/2016	ROUGH RUNNING ENG ON START. ENGINE RAN SMOOTHER AS IT WAR
221646	7230	4/20/2016	DURING IDLE CHECK ON RUN UP, ENGINE QUIT. RESTART ENGINE & Q

Table 1: Five sample of Maintnet's aviation dataset.

Standard NLP tools, however, are typically trained on standard contemporary corpora (e.g. newspaper texts). They struggle when dealing with the domain specific terminology, abbreviations, and non-standard spelling which are abundant in maintenance records. To help encourage further study in this

Proceedings of the 27th International Conference on Computational Linguistics, pages 7–11
Barcelona, Spain (Online), December 12, 2020.

area, we present MaintNet[1], a collaborative, open-source library for technical language resources with a special focus on predictive maintenance data.

The main contributions of this paper are the following:

1. The development of MaintNet, a user-friendly web-based platform that serves as a repository hosting a variety of resources and tools developed to process predictive maintenance and technical logbook data.
2. The creation of several important language resources for technical language and predictive maintenance such as abbreviation lists, morphosyntactic information lists, and termbanks for the aviation, automotive, and facility maintenance domains. All these resources as well as raw data from these domains are made freely available to the research community via MaintNet.
3. The development of several novel Python packages for (pre-)processing technical language which we make available to the research community. This includes stop word removal, stemmers, lemmatizers, POS tagging, document clustering, and more.
4. A collaborative environment in which the community can contribute with data and resources and interact with developers and other members of the community via forums.

2 MaintNet Features

2.1 Language Resources

To the best of our knowledge, there are no freely available tools and libraries developed to process such data, which makes MaintNet a unique resource. MaintNet currently features datasets from the aviation, automotive, and facilities domains (see Table 2), and it will be expanded with the collaboration of the interested members of the NLP community working on similar topics.

Domain	Dataset	Instances	Tokens	Source
Automotive	Maintenance	617	4,443	Connecticut Open Data
	Accident	54,367	242,012	NYS Department of Motor Vehicles
Aviation	Maintenance	6,169	76,866	University of North Dakota Aviation Program
	Accident	5,268	162,533	Open Data by Socrata
Facility	Maintenance	87,276	2,469,003	Baltimore City Maryland Preventive Maintenance

Table 2: The number of instances and tokens in each dataset/domain.

Predictive maintenance datasets are hard to obtain due to the sensitive information they contain. Therefore, we work closely with the data providers to ensure that any confidential and sensitive information in the dataset remains anonymous. In addition to the datasets, MaintNet further provides the user with domain specific abbreviation dictionaries, morphosyntactic annotation, and term banks. The abbreviation dictionaries contains abbreviated validated by domain experts. The morphosyntactic annotation contains the part of speech (POS) tag, compound, lemma, and word stems. Finally, the domain term banks contain the collected list of terms that are used in each domain along with a sample of usage in the corpus.

2.2 Pre-processing and Tools

Grouping maintenance issues by time is an important step in the analysis of logbook data. Most of the predictive maintenance datasets available, however, do not feature the reason for maintenance or the category of the issues making it impossible to train classification systems on such systems. To address this problem, we implemented several (pre-)processing steps to clean and extract information from logbooks aiming at document clustering and classification. The complete processing pipeline is shown in Figure 1.

The pre-processing steps start with text normalization, lowercasing, stop word and punctuation removal. Then we treat special characters with NLTK's (Bird et al., 2009) regular expression library,

[1] Available at: `https://people.rit.edu/fa3019/MaintNet/`

followed by stemming (Snowball Stemmer), lemmatization (WordNet (Miller, 1992)), and tokenization (NLTK tokenizer). POS annotation is carried out using the NLTK POS tagger. Finally, Term frequency-inverse document frequency (TF-IDF) is obtained using the *gensim tfidf model* (Rehurek and Sojka, 2010).

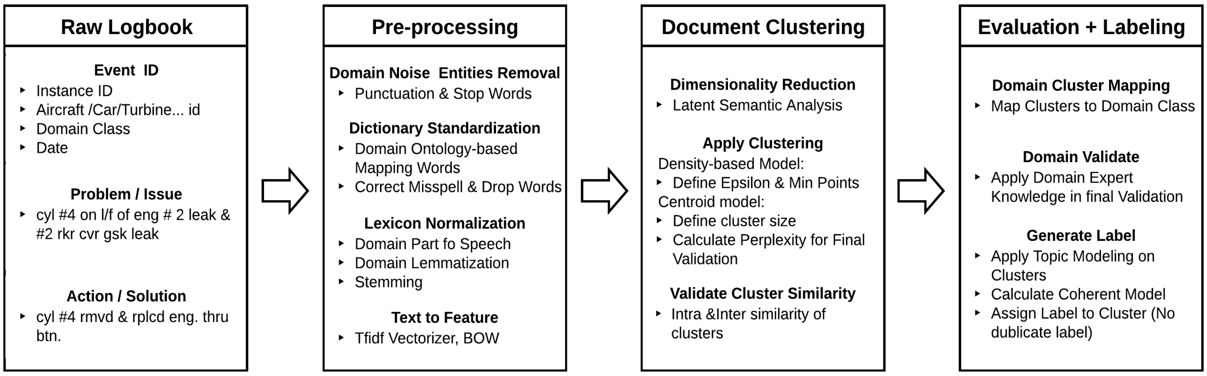

Raw Logbook
Event ID
Instance ID
Aircraft /Car/Turbine... id
Domain Class
Date
Problem / Issue
cyl #4 on l/f of eng # 2 leak & #2 rkr cvr gsk leak
Action / Solution
cyl #4 rmvd & rplcd eng. thru btn.
Pre-processing
Domain Noise Entities Removal
Punctuation & Stop Words
Dictionary Standardization
Domain Ontology-based Mapping Words
Correct Misspell & Drop Words
Lexicon Normalization
Domain Part fo Speech
Domain Lemmatization
Stemming
Text to Feature
Tfidf Vectorizer, BOW
Document Clustering
Dimensionality Reduction
Latent Semantic Analysis
Apply Clustering
Density-based Model:
Define Epsilon & Min Points
Centroid model:
Define cluster size
Calculate Perplexity for Final Validation
Validate Cluster Similarity
Intra &Inter similarity of clusters
Evaluation + Labeling
Domain Cluster Mapping
Map Clusters to Domain Class
Domain Validate
Apply Domain Expert Knowledge in final Validation
Generate Label
Apply Topic Modeling on Clusters
Calculate Coherent Model
Assign Label to Cluster (No dublicate label)

Figure 1: A pipeline of pre-processing and information extraction of maintenance dataset in MaintNet.

To address misspellings and abbreviations which are abundant in predictive maintenance datasets, we explored various state-of-the-art spellcheckers including Enchant[2], Pyspellchecker[3], Symspellpy[4], and Autocorrect[5]. We also developed our own spell checker using Levenshtein distance (Aggarwal and Zhai, 2012) where a dictionary of domain specific words is used to map the misspelling candidates to words in the dictionary. The Levenshtein algorithm was chosen over other distance metrics (*e.g.*, Euclidian, Cosine) as it allows us to control the minimum number of string edits. The performance of our method compared to other spellcheckers in a sub set of the aviation dataset is presented in Table 3.

Total Number of Documents		500
Tokens		3299
Non-standard		289
Success Rate (%)	Enchant	84
	PySpellchecker	61
	Autocorrect	73
	Levenshtein	98

Table 3: Results of the spelling correction and abbreviation expansion methods in terms of success rate.

In MaintNet we also developed document clustering systems customized to logbook data and we make the scripts available to the community. As previously stated, logbook datasets are often not annotated with issue categories requiring a domain expert to group instances into categories. Here we use clustering methods to help grouping documents together.

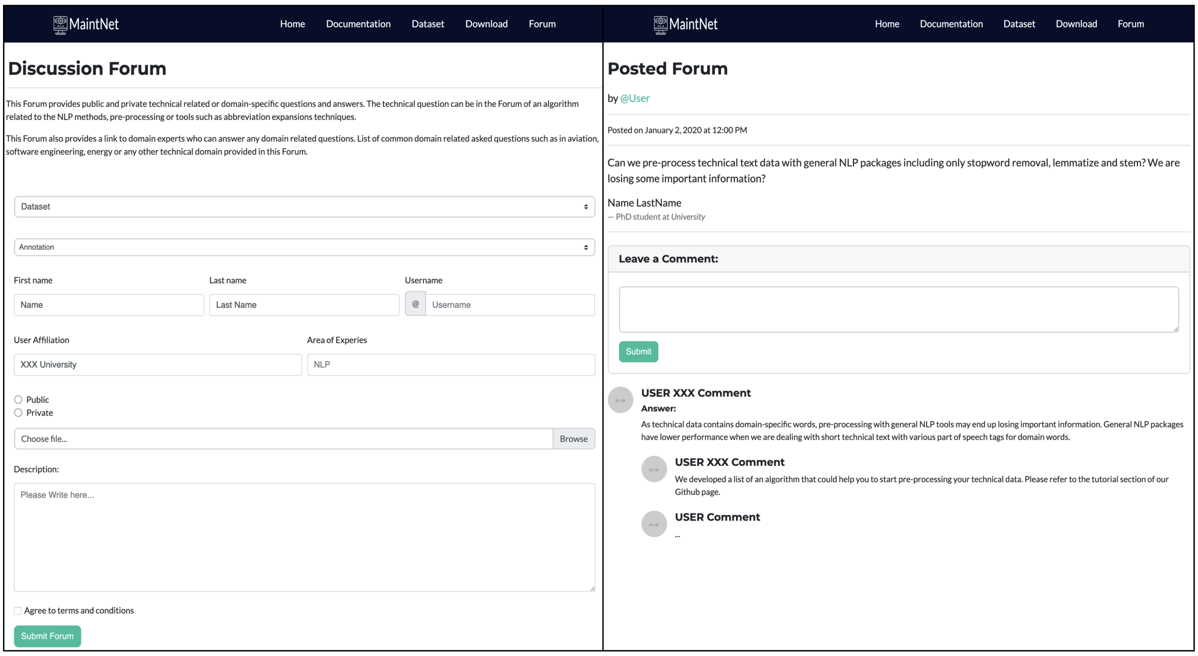
We first convert tokens into a numerical representation using *tfidfvectorizer* (ElSahar et al., 2017) and we obtain a large matrix of document terms (DT). We used truncated singular value decomposition (SVD) (ElSahar et al., 2017) known as latent semantic analysis (LSA), to perform dimensionality reduction. We then experimented with four clustering techniques: k-means (Jain, 2010), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996), Latent Dirichlet Analysis (LDA) (Vorontsov et al., 2015), and hierarchical clustering (Aggarwal and Zhai, 2012). DBSCAN and hierarchical clustering do not require a predetermined number of clusters. For k-means, silhouette and inertia (Fraley and Raftery, 1998) were used to determine the number of clusters while perplexity (Fraley and Raftery, 1998) and coherence (Vorontsov et al., 2015) scores were used for LDA.

[2] `https://www.abisource.com/projects/enchant/`
[3] `https://github.com/barrust/pyspellchecker`
[4] `https://github.com/wolfgarbe/SymSpell`
[5] `https://github.com/fsondej/autocorrect`

Finally, we use three different similarity algorithms: Levenshtein, Jaro, and cosine (Fraley and Raftery, 1998) to calculate intra- and inter-cluster similarity. Cosine similarity is commonly used and is independent of the length of document, while Jaro is more flexible by providing a rating of matching strings. We collected human annotated instances by a domain expert to serve as our gold standard, and these are provided on MaintNet to encourage research into improving unsupervised clustering of maintenance logbooks.

2.3 Community Participation

MaintNet provides various webpages for users to communicate with each other and the project developers; as well as upload data to share with the community (see Figure 2). We hope this will help further facilitate discussion and research in this important and under explored area.



Discussion Forum

This Forum provides public and private technical related or domain-specific questions and answers. The technical question can be in the Forum of an algorithm related to the NLP methods, pre-processing or tools such as abbreviation expansions techniques.

This Forum also provides a link to domain experts who can answer any domain related questions. List of common domain related asked questions such as in aviation, software engineering, energy or any other technical domain provided in this Forum.

Dataset

Annotation

First name: Name
Last name: Last Name
Username: @ Username

User Affiliation: XXX University
Area of Experies: NLP

○ Public
○ Private

Choose file... Browse

Description:
Please Write here...

□ Agree to terms and conditions

Submit Forum

Posted Forum

by @User

Posted on January 2, 2020 at 12:00 PM

Can we pre-process technical text data with general NLP packages including only stopword removal, lemmatize and stem? We are losing some important information?

Name LastName
— PhD student at *University*

Leave a Comment:

Submit

USER XXX Comment
Answer:
As technical data contains domain-specific words, pre-processing with general NLP tools may end up losing important information. General NLP packages have lower performance when we are dealing with short technical text with various part of speech tags for domain words.

USER XXX Comment
We developed a list of an algorithm that could help you to start pre-processing your technical data. Please refer to the tutorial section of our Github page.

USER Comment
...

Figure 2: A screenshot of MaintNet's discussion webpages.

3 Conclusions and Future Work

In this paper we presented MaintNet, a collaborative open-source library for predictive maintenance language resources. MaintNet provides raw technical logbook data as well as several language resources such as abbreviation lists, morphosyntactic information lists, and termbanks from the aviation, automotive and facilities domains. Tools developed in Python are also made available for pre-processing, such as spell checking, POS tagging, and document clustering. In addition to these tools, the collaborative aspects of MaintNet should be emphasized. We welcome the community to contribute with new datasets that can be processed using the tools available at MaintNet, or share new and improved tools developed with MaintNet's open source data.

MaintNet is also expanding as current work involves processing data from additional domains such as healthcare and power systems (*e.g.*, wind turbines). These datasets will be made available on MaintNet in upcoming months. We also aim to collect and release datasets and tools for languages other than English in the near future.

Acknowledgments

We would like to thank Rachael Thormann for the voiceover video. We further thank the University of North Dakota aviation program for the aviation maintenance records dataset and Zechariah Morgain for evaluating the results of the pre-processing and clustering algorithms.

References

Charu C. Aggarwal and ChengXiang Zhai. 2012. A survey of text clustering algorithms. In *Mining Text Data*.

M. Tarik Altuncu, Erik Mayer, Sophia N. Yaliraki, and Mauricio Barahona. 2018. From text to topics in healthcare records: An unsupervised graph partitioning methodology. *ArXiv*, abs/1807.02599.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Hady ElSahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frédérique Laforest. 2017. Unsupervised open relation extraction. *ArXiv*, abs/1801.07174.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*.

Chris Fraley and Adrian E. Raftery. 1998. How many clusters? which clustering method? answers via model-based cluster analysis. *Comput. J.*, 41:578–588.

Anil Kumar Jain. 2010. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31:651–666.

Gabriel Jarry, Daniel Delahaye, Florence Nicol, and Eric Feron. 2018. Aircraft atypical approach detection using functional principal component analysis. In *SID*.

George A. Miller. 1992. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *LREC*.

Ludovic Tanguy, Nikola Tulechki, Assaf Urieli, Eric Hermann, and Céline Raynal. 2016. Natural language processing for aviation safety reports: From classification to interactive analysis. *Computers in Industry*, 78:80–95.

Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. 2015. Bigartm: Open source library for regularized multimodal topic modeling of large collections. In *AIST*.