TODO TITLE

A report submitted to the University of Manchester for the degree of Bachelor of Science in the Faculty of Science and Engineering

Author: Sai Putravu
Student id: 10829976
Supervisor: TODO

2025

School of Computer Science

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

Alphabetically sort this

CBM	Condition-based maintenance policy.
CNN	Convolutional Neural Network.
DL	Deep Learning.
FLD	First-line diagnosis system.
IoT	Internet of Things.
ML	Machine Learning.
PdM	Predictive maintenance policy.
PvM	Preventative maintenance policy.
RF	Random Forest.
R2F	Run-to-failure maintenance policy.
SAFE	Supervised Aggregative Feature Extraction.
SVM	Support Vector Machine.
HDD	Hard-Disk Drive.
SMART	Self-monitoring and reporting technology.
NLP	Natural Language Processing.
LSA	Latent Semantic Analysis.
SVD	Singular Value Decomposition.
ELMo	Embeddings from Language Models.
GPT	Generative Pre-trained Transformer.
BERT	Bidirectional Encoder Representation from Transformers.
UMAP	Uniform Manifold Approximation and Projection.
PCA	Principle Component Analysis.
t-SNE	t-distributed Stochastic Neighbour Embedding.
DBSCAN	Density-Based Spatial Clustering of Application with Noise.
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Application with Noise.

Abstract

Chapter 1

Introduction

: Introduce the sections of the paper.

- Section 2,
- Section 3,
- ...

1.1 Background

: Introduce the research topic. The things in this section will include

- Talk about the ISIS research facility
- Talk about the Operational Cycle for ISIS (graph too)
- Talk about the ISIS Crew and importance of having trained staff on premises.
- Talk about the Lost time and why it is important to minimise this for the ISIS research facility.
- Describe the first-line diagnosis system (FLD) and FAPs.
- Talk about the Datasets, operalog

1.2 Motivation

Motivate the research project.

: The things in this section will include

- Introduce the problem: Auto-categorisation and label inference
- Identify the data input, expected output, data shape and explain why this motivates the project
- Natural Language Processing
- Semantic Similarity
- Need for clustering

Chapter 2

Literature Review

: The things in this section will include

- Looking at general predictive maintenance
- Looking at general predictive maintenance in industrial applications
- Similar pairwise sentence similarity literature
- Similar literature in text clustering
- Similar literature in specifically sentence clustering in industrial applications

2.1 Maintenance Techniques

In the industry, the uptime of production systems are strongly coupled with the equipment maintenance. So much so that what was once considered a "necessary evil" is now seen as a "profit contributor" to be able to maintain a world-class competitive edge [41, 12]. For research facilities providing free-to-use systems, maintenance impacts the downtime and cost of running. As a result, both to minimise unexpected downtime and provide a world-class competitive edge, many industrial applications collect vast quantities of metrics during the entire life cycle of the system. This large amount of data may include information about processes, events and alarms [8] which occur along the industrial production line, collected by different equipment. These equipment may be located in different locations in the sub-components of the larger system or even different sub-components themselves.

talk about maintenance itself in a lot more depth

In literature, various terms and categories of maintenance arise each with differing strategies. Thus, while there exists some disagreement in nomenclature, we consider the categories presented in [37]. The four maintenance policy categories are as follows, noting that each policy has, uniquely, their own benefits and drawbacks:

1. Run-to-failure (R2F) maintenance: Continual usage of the system until failure. Restoration is performed at the point of noticing failure condition. The simplest approach and typically the most costly method, due to requiring an accumulation of a large amount

of defective components which require replacement as well as the consequentially large amount of necessary downtime.

2. Preventative maintenance (PvM): Otherwise referred to as scheduled maintenance, applying maintenance at regular intervals in anticipation for failure of components. While this typically prevents many errors, it wastes maintenance cycles when systems are perfectly healthy. Hence, causing unnecessary downtime and cost.

3. Condition-based maintenance (CBM): Taking the action to perform maintenance on equipment through monitoring various health characteristics and metrics of the components of the system. This approach requires continuous monitoring and, thus, allows for close to instant response on maintenance only when required. However, a drawback of this policy is that one cannot plan maintenances in advance.

4. Predictive maintenance (PdM): Otherwise referred to as statistical-based maintenance, only performs maintenance actions when determined necessary. Prediction tools are utilised to implement forward-planning and scheduling systems, using statistical inference methods. However, if these statistical inferences are not accurate, the whole system suffers which inevitably leads to additional downtime and costs.

It should be noted, that several sources conflate CBM and PdM [23]. As in [37], we refer to them as separate categories.

The PdM strategy stands out in the four categories presented as, given a statistical inference model that is able to detect faults efficaciously, this policy optimises the trade-off between improving equipment condition, reduce failure rates for equipment and minimising maintenance costs [8]. This technique enables one to apply foresight for pre-emptive scheduling of large-scale maintenance. As pointed out in Section (...), the ISIS facility aims to strike a balance between PvM, CBM and PdM through periods of large-scaled scheduled maintenance and collection of high quantities of metrics. This is done through the careful coordination between cycle scheduling, day-to-day crew-based monitoring and the FLD [38].

Implement the ISIS version of this in the background. Talk about the FIRST LINE DIAGNOSIS system

In the industry, many maintenance strategies prefer using PdM whilst experimenting with a variety of statistical inference and artificial intelligence modelling approaches [23, 14]. Some examples from [8] are listed in Table 2.1 which highlights the trend in the industry towards more accurate, ML-based approaches.

fill this

2.2 Sentence Similarity

Think of a good transition between PdM and Doc Similarity

Sentence similarity, otherwise referred to as document similarity, is the (NLP) task of computing the quantification of the similarities between two sentences, documents or texts. This task is motivated by the increasingly large amount of digitisation of human languages (and data, in general), calling for the need to understand similarity between various texts [30]. Examples of the use-cases of sentence similarity include: detection of academic malpractice via plagiarism [20, 3] and text summarisation [2, 16, 15]. According to [30], there are two main types of sentence similarities: (1) lexical similarity and (2) semantic similarity. The former is

Table 2.1: Examples of applications of PdM for industrial maintenance strategies.

Reference	Type	Description
[36]	Statistical	Application of SAFE to deal with PdM problems characterised by time-series data. The approach is tested on a real-life dataset of the semiconductor ion implantation process.
[19]	ML	Application of SVM classification for fault prediction of rail networks, with discussion on using the model in optimising trade-offs related to maintenance schedule and costs.
[26]	ML	Audio analysis on IoT devices, enabling acoustic event recognition for machine diagnosis. This paper describes designing an end-to-end system, utilising CNN-based classification.
[35]	ML	Utilisation of RF decision trees trained on SMART data to predict reliability of HDD in real-time.

a computation of the equality between the lexicon of two sentences (i.e. a purely syntactical view), as opposed to the latter being a comparison between the semantics. Further, the type we focus on, semantic similarity can be split into three types:

- String-based similarity: Measures similarity directly between two strings, accounting for string sequences and character composition. These can be fine-grained, i.e. character-based; coarse-grained, i.e. term-based; or a hybrid mixture of both [44].

- Knowledge-based similarity: Measures the degree to which two sentences are related, utilising semantic networks (i.e. knowledge graphs). Examples of Knowledge-based similarity approaches include WordNet [5], the most popular type of approach.

- Corpus-based similarity: Premised on a provided corpus, a large database of text to derive inferences from. Methods of this type require the development statistical or DL models that train on the provided corpus and estimate the similarity between two sentence-pair inputs. Popular examples include traditional statistical models, such as LSA [17] and SVD [34] as well as word embedding models (utilising ML), such as Word2Vec [4], GloVe [28] and fastText [22].

Most of the models mentioned above require some translation of text into a vector-based representation. Thus, the problem of sentence similarity can be directly mapped from the problem of sentence embedding (otherwise referred to as text embedding) - learning a higher-dimensional embedding space representation. Moreover, with the advent of the transformer architecture [40] and rise of the large language models, text embedding has been increasingly solved using DL models with high parameter counts [7] - with the word embedding models, described previously, only being considered second-generation. Further, according to [7], newer generations fall into the following categories:

- Third-generation: contextualised embeddings. These models dynamically account for contexts, encoding them into the embedding space. Examples of models include ELMo [32], GPT [29] and BERT [10].

- Fourth-generation: universal text embeddings. The generation which is currently state-of-the-art, with the aim of developing a unified model which is able to address multiple downstream tasks. Examples of models in this generation, making progress towards unification include Gecko [18], Multilingual e5 text embeddings [42], Nomic [25] and many more.

Second-, third- and fourth-generation text embedding models are used frequently in PdM for applications such as insight extraction [1, 39] and clustering intents from unstructured text data [24]. Sources of natural language datasets, in industrial applications typically arise from operational or managerial log files which document aspects such as failures, resolutions and comments similar to the ISIS facility failure logs ([**?**] see Section **??**). Advanced text embedding models enable for semi- or fully automatic insight retrieval and auto-categorisation, enabling intuitive understanding of the textual datasets potentially highlighting patterns in failure.

cite and fill when section exists

2.3 (maybe) Clustering

Talk about clustering lit. rev.

Think whether it is useful to present literature review in this section.

Think of a good transition between Sentence Similarity and Clustering

: The things in this section will include

- (DONE) Looking at general predictive maintenance
- (Done) Looking at general predictive maintenance in industrial applications
- (Done) Similar pairwise sentence similarity literature
- Similar literature in text clustering
- (sort of DONE) Similar literature in specifically sentence clustering in industrial applications

Chapter 3

Technical Background

This chapter delves into the technical background required in understanding and appreciating the approach proposed in Chapter 4. Firstly, in Section 3.1.1, we discuss the technical details of the (third-generation text embedding) BERT model and its family of encoder-only transformers [10]. Specifically we further explore two improvements over BERT (XLNet [43] and MPNet [33]). Then, in Section 3.1.2, we explore the state-of-the-art, fourth-generation Nomic [25] architecture. After, we cover two methods of dimensionality reduction (PCA [27, 13] and UMAP [21]) motivated by the need to visualise samples from the high-dimensional embedding spaces of the aforementioned models, in Section 3.2. Finally, we present three clustering algorithms - with one supervised (k-Medoids []) and two unsupervised (DBSCAN [11] and HDBSCAN [6]) in addition to four clustering evaluation metrics. The clustering metrics we look at are: (1) Inertia [], (2) Silhouette [31], (3) Davies-Bouldin Index [9], (4) Calinski-Harabasz Index [**?**].

find the reference for this

maybe talk about optuna, if we use it.

double check all citations here are not empty

: Describe the various technical factors required before attempting to understand the methodology.
The things in this section will include

- Discuss sentence embedding, similarity measures: BERT, RoBERTA, MPNet, XLNet, NOMIC.
- Dimensional reduction techniques and need for them (UMAP, PCA, t-SNE).
- Clustering methods: kmedoids, DBSCAN, DBSCAN*/HDBCAN
- Clustering evaluation methods:
- Maybe briefly touch on Optuna?

3.1 Sentence Embedding

3.1.1 BERT-family Transformers

BERT

XLNet

MPNet

3.1.2 Nomic

3.2 Dimensionality Reduction

3.2.1 PCA

3.2.2 UMAP

3.3 Clustering

3.3.1 k-Medoids

3.3.2 DBSCAN

3.3.3 HDBSCAN

3.4 Clustering Evaluation

3.4.1 Inertia

3.4.2 Silhouette

3.4.3 Davies-Bouldin Index

3.4.4 Calinski-Harabasz Index

3.5 (Maybe) Optuna

Chapter 4

Methodology

: Describe the methods and procedures used. The things in this section will include.

- Explaining data format and data visualisation: wordcloud.
- Data cleaning steps, including removing key words such as Ion Source.
- Text preprocessing steps (cleaning) and computational challenges (tensorflow).
- Choosing the best sentence embedding transformer: MPNET, NOMIC.
- Data visualisation (before and after sentence embedding): similarity visualisation, explain unique sentences, token length distribution.
- Motivate why clustering in higher dimensions performs worse
- UMAP, PCA, t-SNE comparison. Motivate using UMAP.
- UMAP hyperparameter optimisation.
- Performing clustering with kmedoids, dbscan, hdbscan.
- Using optuna.
- Evaluation of results and choosing the best model (and arguing why hdbscan is the best by looking at the variance of dbscan and inflexibility of kmedoids)
- Touch on the production of a CLI application that allows you to mix and match various parts of the pipeline. Motivate the need for command line tool.

Chapter 5

Results and Discussion

: Describe the results and analyse the results

- Analyse the word cloud.
- Analyse the sentence embedding results.
- Analyse UMAP vs. PCA vs. t-SNE qualitatively and later quantitatively (compared to the clustering).
- Anaylse the UMAP hyperparameter optimisation qualitatively, mention that we use Optuna.

Chapter 6

Conclusion

: Summarize your findings and suggest areas for future work.

Bibliography

[1] PY Abijith, Piyush Patidar, Gaurav Nair, and Rohan Pandya. Large language models trained on equipment maintenance text. In *Abu Dhabi International Petroleum Exhibition and Conference*, page D021S065R003. SPE, 2023.

[2] Ramiz M Aliguliyev. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, 36(4):7764–7772, 2009.

[3] Kensuke Baba, Tetsuya Nakatoh, and Toshiro Minami. Plagiarism detection using document similarity based on distributed representation. *Procedia computer science*, 111:382–387, 2017.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[5] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*, volume 2, pages 2–2, 2001.

[6] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[7] Hongliu Cao. Recent advances in text embedding: A comprehensive review of top-performing methods on the mteb benchmark. *arXiv preprint arXiv:2406.01607*, 2024.

[8] Thyago P Carvalho, Fabrízzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

[9] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the*

2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pages 4171–4186, 2019.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[12] Maurizio Faccio, Alessandro Persona, Fabio Sgarbossa, and Giorgia Zanin. Industrial maintenance policy development: A quantitative framework. *International Journal of Production Economics*, 147:85–93, 2014.

[13] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[14] Ali Jezzini, Mohammad Ayache, Lina Elkhansa, Bassem Makki, and Maya Zein. Effects of predictive maintenance (pdm), proactive maintenace (pom) & preventive maintenance (pm) on minimizing the faults in medical instruments. In *2013 2nd International conference on advances in biomedical engineering*, pages 53–56. IEEE, 2013.

[15] Taeho Jo. K nearest neighbor for text summarization using feature similarity. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pages 1–5. IEEE, 2017.

[16] Sushil Kumar and Komal Kumar Bhatia. Semantic similarity and text summarization based novelty detection. *SN Applied Sciences*, 2(3):332, 2020.

[17] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[18] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024.

[19] Hongfei Li, Dhaivat Parikh, Qing He, Buyue Qian, Zhiguo Li, Dongping Fang, and Arun Hampapur. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26, 2014.

[20] Romans Lukashenko, Vita Graudina, and Janis Grundspenkis. Computer-based plagiarism detection methods and tools: an overview. In *Proceedings of the 2007 international conference on Computer systems and technologies*, pages 1–6, 2007.

[21] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[23] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[24] Giancarlo Nota, Alberto Postiglione, and Rosario Carvello. Text mining techniques for the management of predictive maintenance. *Procedia Computer Science*, 200:778–792, 2022.

[25] Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

[26] Zhaotai Pan, Yi Ge, Yu Chen Zhou, Jing Chang Huang, Yu Ling Zheng, Ning Zhang, Xiao Xing Liang, Peng Gao, Guan Qun Zhang, Qingyan Wang, et al. Cognitive acoustic analytics service for internet of things. In *2017 IEEE International Conference on Cognitive Computing (ICCC)*, pages 96–103. IEEE, 2017.

[27] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[28] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[29] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[30] T Nora Raju, PA Rahana, Raichel Moncy, Sreedarsana Ajay, and Sindhya K Nambiar. Sentence similarity-a state of art approaches. In *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, pages 1–6. IEEE, 2022.

[31] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[32] Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

[33] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

[34] Josef Steinberger and Karel Ježek. Text summarization and singular value decomposition. In *Advances in Information Systems: Third International Conference, ADVIS 2004, Izmir, Turkey, October 20-22, 2004. Proceedings 3*, pages 245–254. Springer, 2005.

[35] Chuan-Jun Su and Shi-Feng Huang. Real-time big data analytics for hard disk drive predictive maintenance. *Computers & Electrical Engineering*, 71:93–101, 2018.

[36] Gian Antonio Susto and Alessandro Beghi. Dealing with time-series data in predictive maintenance problems. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2016.

[37] Gian Antonio Susto, Alessandro Beghi, and Cristina De Luca. A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. *IEEE Transactions on Semiconductor Manufacturing*, 25(4):638–649, 2012.

[38] JWG Thomason. The isis spallation neutron and muon source—the first thirty-three years. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 917:61–67, 2019.

[39] Juan Pablo Usuga-Cadavid, Samir Lamouri, Bernard Grabot, and Arnaud Fortin. Using deep learning to value free-form text data for predictive maintenance. *International Journal of Production Research*, 60(14):4548–4575, 2022.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[41] Geert Waeyenbergh and Liliane Pintelon. A framework for maintenance concept development. *International journal of production economics*, 77(3):299–313, 2002.

[42] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[44] Minghe Yu, Guoliang Li, Dong Deng, and Jianhua Feng. String similarity search and join: a survey. *Frontiers of Computer Science*, 10:399–417, 2016.