

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	iii
List of Tables	iv
Abbreviations and Acronyms	v
1 Introduction	2
1.1 Motivation	2
2 Background	4
2.1 Overview of the ISIS Research Facility	4
2.2 Maintenance Techniques	5
2.3 Sentence Similarity	6
2.4 (maybe) Clustering	8
3 Natural Language Processing Background	9
3.1 Sentence Embedding	10
3.1.1 BERT-family Transformers	10
3.2 Dimensionality Reduction	12
3.2.1 PCA	12
3.2.2 UMAP	12
3.3 Clustering	12
3.3.1 k-Medoids	12
3.3.2 DBSCAN	12
3.3.3 HDBSCAN	12
3.4 Clustering Evaluation	12
3.4.1 Inertia	12
3.4.2 Silhouette	12
3.4.3 Davies-Bouldin Index	12
3.4.4 Calinski-Harabasz Index	12
3.5 Maybe Optuna	12
4 Automatic Label Generation	13
5 Results and Discussion	14

6 Conclusion	15
Bibliography	16

List of Figures

3.1	The overall pre-training and fine-tuning phases for BERT. For different downstream tasks, notice the same architecture, except output layer, is used. Source: (Devlin et al. [11]).	11
-----	---	----

List of Tables

2.1	Examples of applications of PdM for industrial maintenance strategies.	6
-----	--	---

Abbreviations and Acronyms

Alphabetically
sort this

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

Abstract

TODO

Chapter 1

Introduction

This project concerns the use of statistical and Machine Learning models to augment the predictive maintenance process at the STFC Rutherford Appleton Laboratory's ISIS Neutron and Muon research facility. The ISIS Neutron and Muon facility is a research centre for physical and life sciences, owned and operated by the STFC, a council that forms the UK Research and Innovation. In order to produce beams of neutrons and muons, allowing scientists to study materials at the atomic level, large and complex structures and machinery are required. The facility has a wealth of instrumentation taking multitudes of measurements to ensure that proper maintenance is completed in a timely manner.

Finish this off

: Introduce the sections of the paper.

- Refine this after coming back
- Section 2,
- Section 3,
- ...

1.1 Motivation

Motivate the research project.

: The things in this section will include

- Introduce the problem: Auto-categorisation and label inference
- Highlight the research aims. This should highlight the vastly open nature of the project and then hone in on the particular issue I am tackling. (i.e. to progress the state of PdM ...)
- Identify the data input, expected output, data shape and explain why this motivates the project

Chapter 2

Background

: The things in this section will include

- Background
- Literature Review
- 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 Overview of the ISIS Research Facility

: 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

2.2 Maintenance Techniques

In 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 competitive edge [48, 14]. For facilities aiming to provide systems for research, maintenance impacts the downtime and cost of running. As a result, both to minimise unexpected downtime and provide a competitive edge, many industrial applications collect vast quantities of data during the entire life cycle of the system. This large amount of data may include information about processes, events and alarms [9] which occur along the industrial production line, collected by different equipment. The equipment may be located in different locations in the sub-components of the larger system or even different sub-components themselves.

In the literature, various terms and categories of maintenance arise each with differing strategies [42, 27, 41]. Thus, while there exists some disagreement in nomenclature, we consider the four categories presented in (Susto et al. [42]). The four maintenance policy categories are as follows, noting that each policy has, uniquely, their own benefits and drawbacks:

talk about maintenance itself in a lot more depth

1. Run-to-failure (R2F) maintenance: Continual usage of the system until failure. Restoration is performed at the point of noticing failure condition. This is the simplest approach and typically the most costly method as it reduces the facility’s availability and requires a complete replacement of parts.
2. Preventative maintenance (PvM): Otherwise referred to as scheduled maintenance, performing maintenance at regular intervals to increase longevity of the component or in anticipation of the end of expected life of the component. 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 [27]. As in (Susto et al. [42]), where they are given as separate categories, we follow suit.

The PdM strategy stands out in the four categories presented as, given a statistical inference model that is able to detect faults accurately, this policy optimises the trade-off between

improving equipment condition, reduce failure rates for equipment and minimising maintenance costs [9]. 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 [45].

In industry, many maintenance strategies prefer using PdM whilst experimenting with a variety of statistical inference and artificial intelligence modelling approaches [27, 16]. Some examples from [9] are listed in Table 2.1 which highlights the trend in the industry towards more accurate, ML-based approaches.

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

Type	Description	Reference
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.	(Susto and Beghi [41])
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.	(Li et al. [21])
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.	(Pan et al. [30])
ML	Utilisation of RF decision trees trained on SMART data to predict reliability of HDD in real-time.	(Su and Huang [40])

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

fill this

2.3 Sentence 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 [34]. Examples of the use-cases of sentence similarity include: detection of academic malpractice via plagiarism [23, 3] and text summarisation [2, 18, 17]. According to [34], there are two main types of sentence similarities: (1) lexical similarity and (2) semantic similarity. The former is 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:

Think of a good transition between PdM and Doc Similarity

- 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 [52].
- 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 [19] and SVD [39] as well as word embedding models (utilising ML), such as Word2Vec [4], GloVe [32] and fastText [25].

Most of the models mentioned above require some numerical representation of the text to be able to apply mathematical procedures for similarity calculation. Computing this representation involves converting unstructured textual data into one or more vectors. Typically, this process includes (1) general pre-processing steps such as stop-word removal, case normalisation, parts-of-speech tagging, lemmatisation, and tokenisation [43]; and (2) applying an embedding model, either to a single token (word embedding) or to a sequence of tokens (sentence embedding). Step (1) can be seen as a feature extraction step applied on the unstructured textual data, where feature extraction is the process of extracting the most useful components of the data [36]. For example, part-of-speech tagging can be seen as introducing non-trivial features to some token through extracting the surrounding context.

This representation is known as an embedding, with the span of the possible vectors referred to as the embedding space. The dimension of the span is termed the embedding dimension. This is an important concept because the characteristics of the embedding space influence the model’s ability to capture syntactic and semantic meaning in text, as the embedding space itself encodes this information. This can be seen in the Word2Vec model, described in (Bojanowski et al. [4]), which shows different embedding dimensions produce different results. Another conclusion that can be drawn from this paper is that, if embedding space is not constructed to maximise the meaning of texts, the accuracy of model predictions tends to deteriorate.

Therefore, the problem of sentence similarity can be directly mapped from the problem of sentence embedding (otherwise referred to as text embedding), where text embedding is the (NLP) task of learning a high-dimensional embedding space representation. Various aspects of text embedding are more thoroughly covered in Section 3.1. However, with the advent of the transformer architecture [47] and rise of the large language models, text embedding has been increasingly solved using DL models with high parameter counts [8] and considering extremely large token sequences. Nowadays, word embedding models are considered obsolete with (Cao [8]) only considering these models second-generation. Further, the paper states 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 [37], GPT [33] and BERT [11]. As these models are trained to both understand some embedding space and generate natural language text, they are canonically referred to as language models.
- **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 [20], Multilingual e5 text embeddings [49], Nomic [29] and many more.

Second-, third- and fourth-generation text embedding models are used frequently in PdM for applications such as insight extraction [1, 46] and clustering intents from unstructured text data [28]. 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

2.4 (maybe) Clustering

Talk about clustering lit. rev.

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

: 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

Think of a
good transi-
tion between
Sentence Sim-
ilarity and
Clustering

Chapter 3

Natural Language Processing 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 [11]. Specifically we further explore two improvements over BERT (XLNet [51] and MPNet [38]). Then, in Section 3.1.1, we explore the state-of-the-art, fourth-generation Nomic [29] architecture. After, we cover two methods of dimensionality reduction (PCA [31, 15] and UMAP [24]) 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 [13] and HDBSCAN [7]) in addition to four clustering evaluation metrics. The clustering metrics we look at are: (1) Inertia [], (2) Silhouette [35], (3) Davies-Bouldin Index [10], (4) Calinski-Harabasz Index [6].

: 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*/HDBSCAN
- Clustering evaluation methods: todo I don't remember these off the top of my head
- Maybe briefly touch on Optuna?

find the reference for this

maybe talk about optuna, if we use it.

double check all citations here are not empty

3.1 Sentence Embedding

Briefly touched on in Section 2.3, sentence embedding (otherwise known as text embedding) is the NLP task of computing some high-dimensional embedding vector-space representation for unstructured text data. This numerical representation should encode the semantic and syntactic meaning of the text and establish meaningful relationships. For example, the sentence ‘I like dogs’ should have the opposite representation to ‘I hate dogs’. However these sentences should be more related than to the sentence ‘My house was destroyed in an earthquake’. Deep learning sentence embedding models are now seen to be state-of-the-art as they are able to extract features automatically and more effectively than manual efforts, when supported with large quantities of data [22].

Maybe add this as an image illustration

3.1.1 BERT-family Transformers

Pre-trained language models (PLMs), such as BERT [11], have been widely successful in a wide range of NLP tasks through fine-tuning [12, 26]. Fine-tuning is the process of re-training a PLM on specialised tasks, leveraging the model’s base knowledge, by applying perturbations to the pre-trained model parameters through gradient descent learning algorithms. These language models, trained on (1) finding an embedding space for natural language and (2) stochastically generating tokens to mimic natural language, often provide a great platform to perform task-specific model fine-tuning. Platforms such as HuggingFace [50], allow authors to upload these pre-trained model parameters which in-turn allows researchers to download them for them fine-tuning. Moreover, task-specific fine-tuned models parameters are uploaded, downloaded and shared on these platforms. These fine-tuned models are useful for researchers whose focus lies outside of optimising these model parameters. In this section, we talk about the BERT-family of language models.

BERT

The Bidirectional Encoder Representations from Transformer (BERT) language model introduced in (Devlin et al. [11]) was designed to learn deep bidirectional representations from unlabelled text. It achieved this by learning the left and right context in every layer of the model. The model implementation can be split into two phases: (1) the pre-training phase; and (2) the fine-tuning phase. This allows the model architecture to remain common, with many down-stream NLP tasks benefiting from a single PLM. An example illustration can be seen in Figure 3.1, where a single model can be fine-tuned on many down-stream tasks by simply replacing the output layer. The model architecture is a multi-layer Transformer encoder based on the original paper (Vaswani et al. [47]) which is bidirectional by nature. As the transformer architecture is a very well researched architecture, well represented in the literature and slightly out-of-scope for this paper, we refer the reader to the original paper.

sprinkle some images in here.

The pre-training phase can be split into two further training tasks:

- Masked Language Model - During training, a portion of the input tokens are ‘masked’ and the model predicts these masked tokens based on the remaining context, using cross-entropy loss. In the literature, this is referred to as a Cloze task [44]. In the

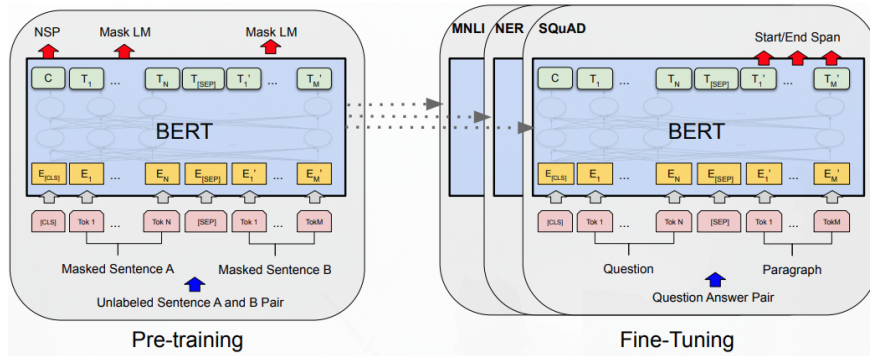


Figure 3.1: The overall pre-training and fine-tuning phases for BERT. For different downstream tasks, notice the same architecture, except output layer, is used. Source: (Devlin et al. [11]).

experiments conducted by (Devlin et al. [11]) 15% of tokens in each sequence are randomly selected for masking. However, to mitigate the mismatch between pre-training and fine-tuning phases, a training data generator replaces each chosen masked tokens with: (1) the special token **[MASK]** 80% of the time; (2) a random token 10% of the time; and (3) the original token itself 10% of the time.

- **Next Sentence Prediction** - In this task, the model is trained on a binary classification task prediction the relationship between sentence pairs. Given two sentences **A** and **B** either (1) sentence **B** is the actual sentence that follows **A** 50% of the time or (2) sentence **B** is randomly selected from the corpus 50% of the time.

XLNet

MPNet

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

Automatic Label Generation

Chapter 5

Results and Discussion

Chapter 6

Conclusion

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] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.
- [7] 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.
- [8] Hongliu Cao. Recent advances in text embedding: A comprehensive review of top-performing methods on the mteb benchmark. *arXiv preprint arXiv:2406.01607*, 2024.
- [9] Thyago P Carvalho, Fabrizzio 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.
- [10] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

- [11] 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.
- [12] Sergey Edunov, Alexei Baevski, and Michael Auli. Pre-trained language model representations for language generation. *arXiv preprint arXiv:1903.09722*, 2019.
- [13] 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.
- [14] 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.
- [15] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- [16] Ali Jezzini, Mohammad Ayache, Lina Elkhansa, Bassem Makki, and Maya Zein. Effects of predictive maintenance (pdm), proactive maintenance (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.
- [17] 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.
- [18] Sushil Kumar and Komal Kumar Bhatia. Semantic similarity and text summarization based novelty detection. *SN Applied Sciences*, 2(3):332, 2020.
- [19] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.
- [20] 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.
- [21] 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.
- [22] Hong Liang, Xiao Sun, Yunlei Sun, and Yuan Gao. Text feature extraction based on deep learning: a review. *EURASIP journal on wireless communications and networking*, 2017:1–12, 2017.

- [23] 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.
- [24] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [26] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.
- [27] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.
- [28] Giancarlo Nota, Alberto Postiglione, and Rosario Carvello. Text mining techniques for the management of predictive maintenance. *Procedia Computer Science*, 200:778–792, 2022.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [34] 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.
- [35] 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.

- [36] Mark Sammons, Christos Christodoulopoulos, Parisa Kordjamshidi, Daniel Khashabi, Vivek Srikumar, and Dan Roth. Edison: Feature extraction for nlp, simplified. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4085–4092, 2016.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] Ayisha Tabassum and Rajendra R Patil. A survey on text pre-processing & feature extraction techniques in natural language processing. *International Research Journal of Engineering and Technology (IRJET)*, 7(06):4864–4867, 2020.
- [44] Wilson L Taylor. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.
- [45] 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.
- [46] 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.
- [47] 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.

- [48] Geert Waeyenbergh and Liliane Pintelon. A framework for maintenance concept development. *International journal of production economics*, 77(3):299–313, 2002.
- [49] 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.
- [50] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.
- [51] 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.
- [52] Minghe Yu, Guoliang Li, Dong Deng, and Jianhua Feng. String similarity search and join: a survey. *Frontiers of Computer Science*, 10:399–417, 2016.