SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Named Entity Recognition of an Oversampled and Preprocessed Manufacturing Data Corpus

Nurul Hannah Mohd Yusof[1,*], Nurul Adilla Mohd Subha[1], Nurulaqilla Khamis[1], Norikhwan Hamzah[2]

[1] Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[2] Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

ARTICLE INFO

Article history:
Received 22 June 2023
Received in revised form 4 September 2023
Accepted 17 October 2023
Available online 25 December 2023

Keywords:
Named Entity Recognition; Hidden Markov Model; Factory Reports

ABSTRACT

In recent manufacturing industry, improving the manufacturing process is of paramount importance. One area that holds great potential for enhancement is the application and manipulation of maintenance data. By effectively leveraging this data, manufacturers can optimize maintenance schedules, leading to increased efficiency, reduced costs, and minimized downtime. However, the challenge lies in handling vast amounts of maintenance data that often come in various formats, making it difficult to extract valuable insights. Without proper analysis, this unprocessed data can result in unforeseen issues, costly disruptions, and extended downtime periods. To overcome this obstacle, modern manufacturing companies are turning to advanced technologies such as language modelling, text classification, machine translation, and Named Entity Recognition (NER). To the best of our knowledge, no investigation has been conducted to assess the impact of text preprocessing on NER performance. Improving the initial stage of NER, such as text preprocessing, can enhance NER performance which leads to the training model's efficiency performance. In this study, Hidden Markov Model (HMM) is employed to improve NER performance by utilizing oversampling and text preprocessing techniques. The study is performed without IOB labelling and consider seven specific entities and the preprocessing text tasks include tokenization, lemmatization, erase punctuation, stop words removal, and elimination of long and short words. As a result, HMM for NER with oversampling and preprocessed text outperformed the one without any of both by 20.10% and 27.59%, respectively, due to consideration of significant classes and words among the entity classes in preprocessed factory reports. This finding highlights the importance of text preprocessing method selection in NER and its capability to optimize maintenance schedule and reduce downtime.

1. Introduction

There is vast amount of untapped valuable knowledge within manufacturing workers' manual production reports. The manufacturing process can reap significant benefits from these types of reports as long the information within the reports is utilized to influence decision-making process [1].

* *Corresponding author.*
E-mail address: nhannahmyusof@gmail.com (Nurul Hannah Mohd Yusof)

https://doi.org/10.37934/araset.36.1.203216

However, due to the overwhelming amount of available report data, manual categorization becomes a stumbling block that hinders the optimum utilization of the information [2]. Data manipulation becomes harder to achieve with report data that encompasses a broad range of manufacturing domains [3]. Therefore, numerous studies have explored various potential solution to this issue including utilization Named Entity Recognition (NER).

NER is an important part in information extraction of Natural Language Processing (NLP) text mining that localizes entity-containing (e.g., people, organizations, locations) spans in a text and classify them into designated categories, which is crucial for system with large datasets [4]. The accuracy and efficiency of downstream NLP operations can be improved by modifying the way NER handle text input in terms of recognising and labelling the entities. There are several applications of NER in manufacturing sectors such as quality control, supply chain management, compliance monitoring, customer service, and predictive maintenance [5-9].

Rule-based NER is a classic approach for entity identification that entails developing a set of rules or patterns to recognise entities in a text. This method is based on the notion of entities categories such as names of people, organizations, or locations have a unique feature that can be identified by a set of predetermined rules. Under certain circumstances such as when the domain is well-defined and naming practises are consistent, this method is favoured in several previous studies as it has the potential to efficiently detect entities in text, particularly in cases where there is a shortage of annotated corpus resources that can be utilized as training data [10-12]. However, for customized rules, the construction and maintenance of the rule-based NER can be time-consuming. The complexity of the rules rises as larger annotated datasets involved which limit the system scalability and flexibility. Nowadays, rule-based NER can be used with other advance technique such as machine learning to improve entity recognition performance.

To overcome the limitation of the abovementioned methods, statistical models can be adopted in NER, such as Conditional Random Fields (CRF) and Hidden Markov Model (HMM) as discussed in these studies, Kim *et al.,* [13], Kumar and Starly [14], Li *et al.,* [15], and Drovo *et al.,* [16], Shrivastava *et al.,* [17], Mo *et al.,* [18], Pasa *et al.,* [19], respectively. The CRF is a kind of discriminative probabilistic model that can be applied to the NER-based problems that require sequence labelling. As a result of the CRF ability to describe intricate relationships between input and output sequences, it is frequently regarded as a powerful and versatile method [14]. However, there are situations where certain labels or events only appear in the training data, which may result in the CRF models being sensitive to class imbalance or rare events. Furthermore, the noise and errors generated from the input data are difficult to be handled which restrict the application of this method to sequence labelling or structured prediction. These problems may impact the accuracy as well as the performance of the developed CRF models.

The HMM model is used in the NER applications to predict the sequential dependencies between words in speech transcripts and forecast the probability of each word being part of a named entity. The model is developed by using a labelled dataset of speech transcripts, where each word is assigned a label that corresponds to the appropriate named entity. While the HMMs may not be at the forefront of technology for this task, they offer a quick and effective solution for the NER in speech transcripts, particularly when the script is fragmented and noisy. Moreover, the HMMs can be readily adapted to work in various domains and languages, making them a valuable tool for the NER in multiple settings. Additionally, the HMM is a versatile and potent probabilistic model that can interpret intricate patterns and correlations in a sequential data, such as text. As a result, the HMM is exceptional for NER tasks that aim to recognise and categorise entities in a text [20].

Besides the CRF and the HMM, deep learning models such as Bidirectional Encoder Representations from Transformers (BERT) and Long Short-Term Memory (LSTM) networks are some

of the most recent advancements in the NER. Transformer-based NER also use deep learning models based on the Transformer architecture, such as the Bidirectional Encoder Representations from Transformers (BERT) or GPT-3, to capture entities in non-sequential processing [21]. The model is trained on a vast quantity of data and can learn to identify links and patterns in text that are challenging to capture using conventional rule-based or statistical approaches. Despite better accuracy and versatility to a variety of applications, transformer-based methods are computationally expensive models due to their numerous parameters and complicated attention mechanisms considered. This prevents their use in real-time applications or on devices with limited resources. Overall, the NER technology is still rapidly developing with various ongoing research to increase the precision and effectiveness of the NER models specifically for manufacturing applications.

The subsequent subsections provide an overview of the common steps involved in the development of NER. The purpose is to highlight the common method used and highlights existing gaps.

1.1 Labelling Domain Entities

Creating a set of labels that match to the various kinds of entities that are pertinent to a given domain is often the first step in labelling named entities in that domain. For instance, if the domain is biomedical text, the labels might list different kinds of genes, proteins, illnesses, cures, and other biomedical items. It is now possible to develop a specialised NER models that can precisely identify the kinds of entities that are relevant in a certain domain by labelling named entities in that domain. This can increase the precision and efficiency of downstream NLP activities like information retrieval, question answering, and sentiment analysis that depend on a precise identification of the named entities. The NER performance is usually limited by due to lack of resources and a small quantity of annotations. Since a specific domain can serve as the theoretical foundation for a NER work, a specific domain entity must be employed to build up the data sharing between various domains and the shared channel model [22].

Building a domain-specific data set is vital for manufacturing NER to achieve high accuracy. To train and test the NER models, a significant amount of manufacturing-related text data must be gathered and annotated. Hence, specific domain entity must be carefully assigned to attain good prediction of the NER.

It is important to keep in mind that depending on the subfield or application within a given domain, the types of relevant named entities can change. For instance, in the medical domain, various named entity types might be pertinent for various subfields like neurology, cardiology, or oncology. Similarly, distinct named entity types in the military domain might be appropriate for various military equipment or operations. In manufacturing, the domain entity usually used the manufacturing feature library as a guideline which has been implemented in CAD/CAM and CNC machine fault diagnosis. Equally important as appropriately defined domain, the impact of oversampling and text preprocessing on the NER can also influence the performance of a training model noticeably.

1.2 Oversampling

When one or more classes are underrepresented in comparison to other classes in a dataset, a technique called oversampling the minority class is employed to compensate for the imbalance. In certain situations, the model may be biased in favour of the majority class, which inevitably causes it to perform poorly when forecasting the minority class [23]. To boost the minority class's

representation in the dataset, oversampling the minority class entails producing additional synthetic or replicated samples of the minority class. Since the bulk of class-common qualities among data instances are naturally favoured by class-balanced data models, traditional data mining methods are unable to identify uncommon classes [24]. A COS-HMM (Content-based Over-Sampling HMM) is trained with a corpus to generate fresh samples that are consistent with the most recent documents in this previous study [25]. The HMM is viewed as a document generator that can create synthetic instances based on the data trained, however, too many synthetic instances can lead to overfitting or introduce noise in the data. In contrast, if the oversampling is used with caution, the model can learn from more examples as a result, which enhances its capacity to spot and generalise trends for the minority class. The more potent model created from this simpler, smoothed dataset is then oversampled using the original sequences. Importantly, this approach does not call for further modifications to the model's parameters or architecture. Subsequently, the given text dataset's quality can be further improved through text preprocessing.

1.3 Text Preprocessing

A given text dataset's quality can be improved through preprocessing, especially for NER. By encoding each word as a feature vector, text preprocessing seeks to break down each document into distinct words [26]. Text documents are used to create transactions. The keyword must be selected using the feature selection method and main text preprocessing procedures in order to index documents. The text document is processed into several tasks, such as tokenization, words, terms, or characteristics in the text preprocessing stage after the input text documents have been reviewed. Then, a vector space with those features and their weights, which are determined by the frequency of each feature in the text document, serves as a representation of this text document in a data representation. Any non-informative elements (stop words, digits, and special characters) will be eliminated. Then, the remaining characteristics is lemmatization or stemming to the base words. In manufacturing, preprocessed NER has been adopted in these studies [4,27]. Various forms of text preprocessing tasks such as lowercasing stop words and morphological affixes removal contributed to the successful recognition of general domain entities [27]. However, the stemming technique usually generates incorrect spelling and meanings which deem lemmatization to be the more precise method as in this former study [26]. Lemmatization simplifies a word to its core meaning and considers a word’s synonyms. In contrast, context plays a vital part in nearly every element of decision-making and communication in the manufacturing industry. Context, as used in the manufacturing sector, refers to the circumstances or setting in which a given manufacturing process, task, or issue takes place. The materials being used, the machinery being used, the people engaged, the safety rules that must be observed, and the overarching objectives of the manufacturing process are just a few examples of context-related elements. Making wise judgements, spotting potential risks and opportunities, and making sure the manufacturing process is as effective and efficient as possible all depend on understanding the context of a specific manufacturing task or issue. In accordance with that, to investigate the impact of text preprocessing on the NER, numerous evaluation metrics of a confusion matrix can be considered as a performance evaluation tool that is frequently used to assess how well machine learning models perform, especially for classification tasks like the NER.

1.4 Evaluation: Confusion Matrix Chart

A classification model's performance can be assessed by using the confusion matrix to determine several performance measures like accuracy, precision, recall, and F1 score. These performance metrics of an algorithm are calculated based on true positive, true negative, false positive (known as Type I error) and false negative (known as Type II error). Prediction summary can be represented as a confusion chart. It displays the percentage of right and wrong predictions made for each class. It facilitates in comprehending classes that are modelled after other classes but are not quite the same. To increase the model's accuracy and efficacy, patterns and trends in the confusion matrix's performance must be analysed.

The total number of observations in each cell are shown in the confusion matrix. The columns of the confusion matrix correspond to the predicted class, and the rows to the true class as can be seen in Figure 1. Correctly and wrongly classified observations are represented by diagonal and off-diagonal cells, respectively.

Precision is a parameter in the confusion matrix that assesses the share of true positive predictions among all positive predictions. In other words, precision evaluates the accuracy of a model's predictions of successful outcomes. By ensuring that only genuinely positive examples are found, a high precision score can lessen the possibility of incorrect alerts and wasteful interventions [27].

Recall, often referred to as sensitivity or true positive rate, is a parameter that quantifies the fraction of true positive predictions among all real positive cases. High recall score can accurately identify a high percentage of positive cases while low recall score shows that the model is missing a lot of positive examples, which can be problematic in situations where false negatives are expensive or have serious repercussions, like in medical diagnosis or security screening [27].

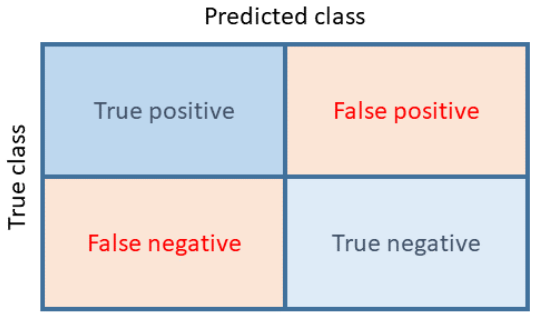

Fig. 1. Confusion matrix chart composition

Precision and recall typically have a trade-off and depending on the needs of a given application, it is frequently required to balance these parameters. For instance, it could be more crucial to have high precision in particular applications, even if this means compromising some recall. Even though this entails accepting a larger proportion of false positives, in other situations, excellent recall might be more crucial.

The harmonic mean of recall and precision in the confusion matrix is represented by the F1-score, which is a single integer. The F1-score establishes a balance between the two metrics by considering both precision and recall. As it accounts for both types of errors (false positives and false negatives) and provides a single number that can be used to compare various models or parameter settings, the F1-score is a useful metric when the number of positive and negative cases in the data is unbalanced [2].

The performance for clean data will eventually increase the reliability of the trained model. In this work, confusion matrix will be used, to analyze the precision, recall, and the F1 score data might be unbalanced due to entity class distribution that will be overviewed in subsection *3.2*, where the distribution of the target variable's observation count across classes is not uniform. To put it another way, some classes have a lot more observations than others.

To the best of our knowledge, there is no comprehensive investigation done on the capability of text preprocessing with lemmatization method in HMM training for producing better NER prediction. A good NER prediction will lead to a more accurate fault diagnosis of machine tools [27]. Hence, this study aims to investigate the NER performance by considering oversampling and text preprocessing technique of HMM. In NLP, lemmatization is a commonly used as a normalization technique due to its high precision in analyzing context of the words to be preprocessed. As a limitation, this study did not consider Inside–outside–beginning (IOB) labelling due to the nature of the available words in factory reports do not require phrases or propositions to be chunked into a unit. Furthermore, the study considers seven entities tags which will be further explained in more detail in Section 2. The difference of general and specific domain entities, and how oversampling and text preprocessing affects the performance of NER using HMM is discussed. Particularly, the NER in this study focusing on specific manufacturing domain entity that is HMM-trained model with processed data of factory reports. A summary of the NER, oversampling, text preprocessing, HMM, and evaluation criteria are provided in Section 1. The procedures are shown in Section 2. Section 3 contains the results and performance evaluation of the research. The conclusion and future directions are narrated in Section 4.

2. Methodology

The experiment in this study was set up by utilizing a 64-bit operating system, an Intel(R) Core(TM) i5-9400F CPU running at 2.90GHz, and 8GB of RAM. Version 21H2 of Windows 10 Pro served as the operating system. There are a few toolboxes that must be installed:

(i) MATLAB R2023a
(ii) Deep Learning Toolbox
(iii) Text Analytics and
(iv) The factory report data set [28]

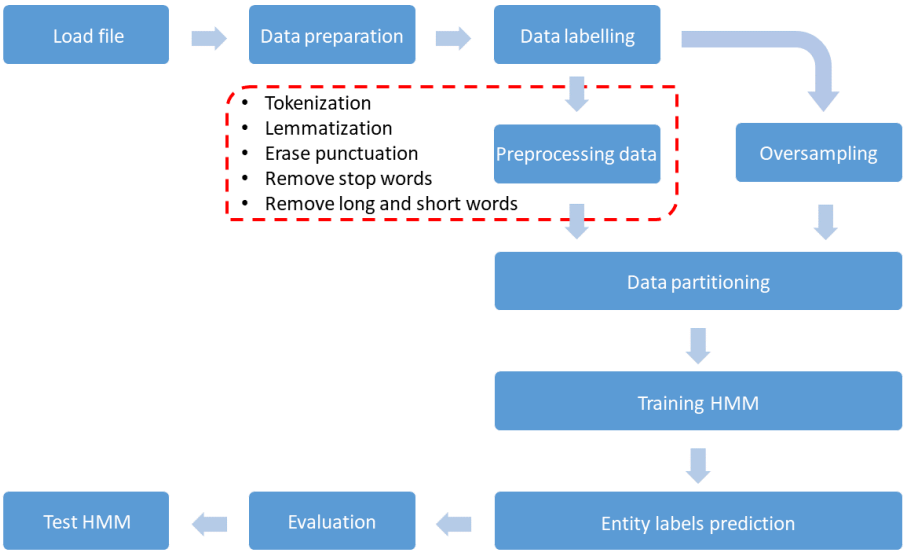

Fig. 2. The steps of HMM construction

By referring to Figure 2, the process of developing HMM started by loading the data file from a comma-separated values (.csv) file from the factory report in MATLAB library [28]. The dataset contains 480 sentences of reports which breaks down into 2724 data. This also involved selecting appropriate corpus or dataset from the factory reports. The data is being tokenized and as a result, every sentence is placed in a cell array. Then, data will be labelled with the entities based on seven specific manufacturing domain entities: “action”, “equipment”, “location”, “material”, “non-entity”, “substance”, and “symptom”. These domain entities are determined based on the manufacturer preference according to the key information extracted from the report.

From the original dataset, to prepare data for oversampling, the data is manually observed to seek for a patch of data that contains highest frequency of small quantity class distributions like “action”, “substance”, and “material” to gain more weightage for these classes. This study chooses 700 data in sequence as the patch data so that the training will be more accurate as in Balakrishnan and Lloyd-Yemoh [23] and Luo *et al.,* [29]. The data is from data number 700 to 1400 which is partitioned into 563 training data and 140 test data.

Text preprocessing, which involved tokenization, lemmatization, removal of punctuation, stop words as well as short and lengthy words, is adopted before data is partitioned into training (80% equivalent to 2180 data) and testing (20% equivalent to 544 data) data set. Then, the trained model of HMM is used to predict the entity of the testing data. The confusion matrix chart is utilized to evaluate the performance of the trained model with and without the text preprocessing. The precision (P), recall (R), and F-1 score ($F1$) are being analyzed to determine the best prediction. Finally, the study will be extended to test the model using a new input text.

A classification model's precision is a parameter in the confusion matrix that assesses the share of true positive predictions among all positive predictions. In other words, precision evaluates the accuracy of a model's predictions of positive outcomes. The equation is defined as

$$P = \frac{TP}{TP+FP} \tag{1}$$

where true positive, TP and false positive, FP respectively. The recall is ratio of TP and false negative, FN. The equation is given by,

$$R = \frac{TP}{TP+FN} \tag{2}$$

F-1 score is the harmonic means between precision and recall which is defined by,

$$F1 = 2 * \frac{P*R}{P+R} \tag{3}$$

The result of this study will be presented in four parts. Part 1, outlined in subsection 3.1, focuses on the comparison between general and specific entities to determine the correctness of the text data. Part 2, detailed in subsection 3.2, provides an overview of the factory reports’ content by presenting the entity class distribution. Part 3, described in subsection 3.3, evaluates the impact of oversampling and text processing on manufacturing NER. Finally, in the last part, outlined in subsection 3.4, the trained HMM is validated using new input text from factory data.

3. Results

This section discusses the results obtained from the NER study. The effects of having specific manufacturing domain entities, oversampling, and text preprocesing in NER data prediction using HMM is discussed in the following subsection.

3.1 Data Preparation and Entities Labelling (General Domain Entities vs Specific Domain Entities)

By referring to Figure 3, general domain entities generated using the MATLAB command *'addEntityDetails'* did not define accurate entity of the word in the factory reports. Most of the words are identified as "non-entity". In contrast, in Figure 4, entity of each word in the factory report are appropriately labelled according to a defined specific manufacturing domain term, which is one of the key pieces of information that are essential to the production process in manufacturing. By effectively managing this information, significant data set can provide a more accurate diagnosis for maintenance which indirectly helps manufacturers to improve efficiency, reduce costs, and ensure that products are manufactured up to the required quality standards.

Token	DocumentNumber	Entity
"Mixer"	1	location
"is"	1	non-entity
"hot"	1	non-entity
"to"	1	non-entity
"the"	1	non-entity
"touch"	1	non-entity
"."	1	non-entity
"Assembler"	2	location

Fig. 3. General domain entities

Token	DocumentNumber	Entity
"Mixer"	1	"equipment"
"is"	1	"non-entity"
"hot"	1	"symptom"
"to"	1	"non-entity"
"the"	1	"non-entity"
"touch"	1	"action"
"."	1	"non-entity"
"Assembler"	2	"equipment"

Fig. 4. Specific manufacturing domain entities

3.2 Entity Class Distribution

After data preparation is completed, the frequency of words according to the entity class distribution is observed to view the overall insight of the factory reports as depicted in Figure 5. Based on the figure, it can be noticed that the entities with high frequency are "equipment", "symptom", and "non-entity" which indicates that these entities are commonly exists in most sentences in the factory report. Minority entities like "action", "location", "material", and "substance" have low frequency of data, however, small miss prediction of these data would certainly cause great impact on performance of the trained model in term of precision. To handle this imbalanced sample, this work has prioritized oversampling techniques to help in balancing the data set and by setting higher

weights for the minority classes, the model pays more attention to these samples during training. This issue will be discussed further in the next subsection.

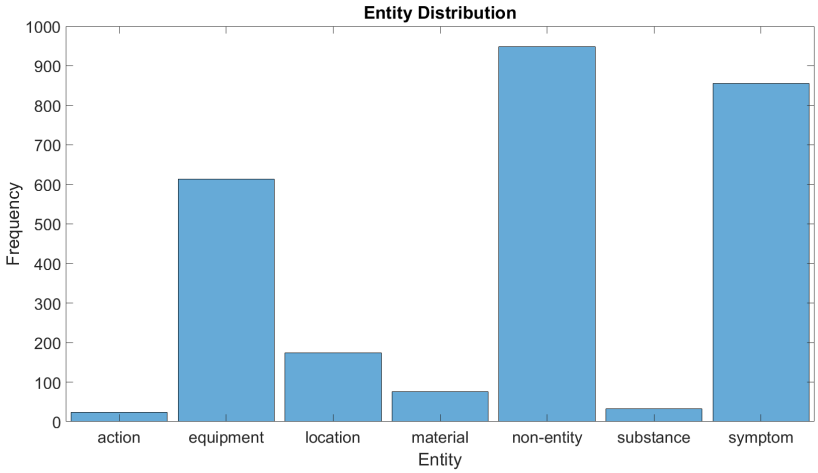

Fig. 5. Entity distribution of factory reports

3.3 Performance Evaluation

From Table 1, precision of the oversampled and preprocessed NER data prediction is better than the one without both except for the "symptom" class. Lower precision in the NER with preprocessed text is observed due to small amount of data falls into "non-entity" class that eliminated the unnecessary information like stop words, punctuation, short, and lengthy words. Even smallest number (2/12 and 1/12) of mistakenly categorised that fell into false negative or false positive would largely affect the precision and recall value.

Table 1
NER precision and recall

Factory reports Entity	Precision (%) (without text preprocessing)	Precision (%) (oversampling)	Precision (%) (with text preprocessing)	Recall (%) (without text preprocessing)	Recall (%) (oversampling)	Recall (%) (with text preprocessing)
Action	30.0	100.0	100.0	75.0	100.0	100.0
Equipment	84.8	90.9	98.3	91.8	93.8	92.7
Location	73.3	75.0	100.0	97.1	100.0	100.0
Material	63.6	100.0	100.0	93.3	100.0	100.0
Non-entity	97.2	97.9	85.7	92.1	95.9	92.3
Substance	38.5	66.7	100.0	71.4	100.0	100.0
Symptom	97.2	94.9	93.7	80.2	86.0	97.4
Average	69.2	89.3	96.8	85.8	96.5	97.5

Figure 6, Figure 7, and Figure 8 show the percentages of successfully and mistakenly categorised observations for each true class. A row summary is designated as recall and for each predicted class in a column summary is termed as precision.

By referring to Figure 6 and Figure 8, "non-entity" class has highest percentage in precision and recall for NER without preprocessed text compared to NER with processing text. However, this class consists of 180 insignificant information such as the stop words, for e.g., 'is', 'to', and 'the' as well as punctuation '.' from the original report as can be seen in Figure 3. These additional data will consume more time to train the model as the number of data is larger than the preprocessed text. In contrast,

with text processing, the prediction for confusion matrix produces more significant data in class of "symptom" and "equipment" which has meaningful details to the maintenance of the factory reports.

With reference to Table 1, Figure 7, and Figure 8, on average of the overall precision between oversampling and text preprocessing has increased by 7.5%, from 89.3% to 96.9% and the overall recall between both has increased by 1%, from 96.5% to 97.5%, which means that both methods are meaningful to be implemented in enhancing accuracy of the HMM predictions. The number mistakenly categorised of true classes can also be reduced.

In accordance with the information presented in Figure 6 and Figure 7, massive increment between normal NER prediction and oversampled NER prediction can be seen in term of precision and recall, which is 20.1% and 10.7% respectively. By considering both precision and recall measure for the prediction, the overall F1-score of NER prediction with oversampling is 0.8934 and with text preprocessing is 0.9681 while the one without both is 0.6924. This indicates that the oversampled and preprocessed NER data prediction performs better regardless the imbalanced dataset because the weight of the minority classes has been increased during oversampling.

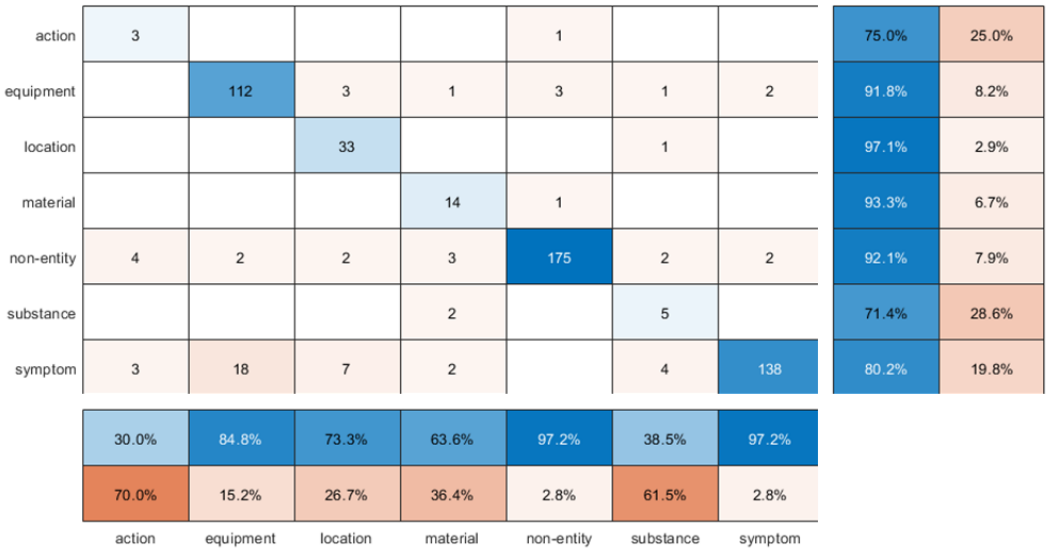

Fig. 6. Confusion chart of NER prediction without oversampling and text preprocessing

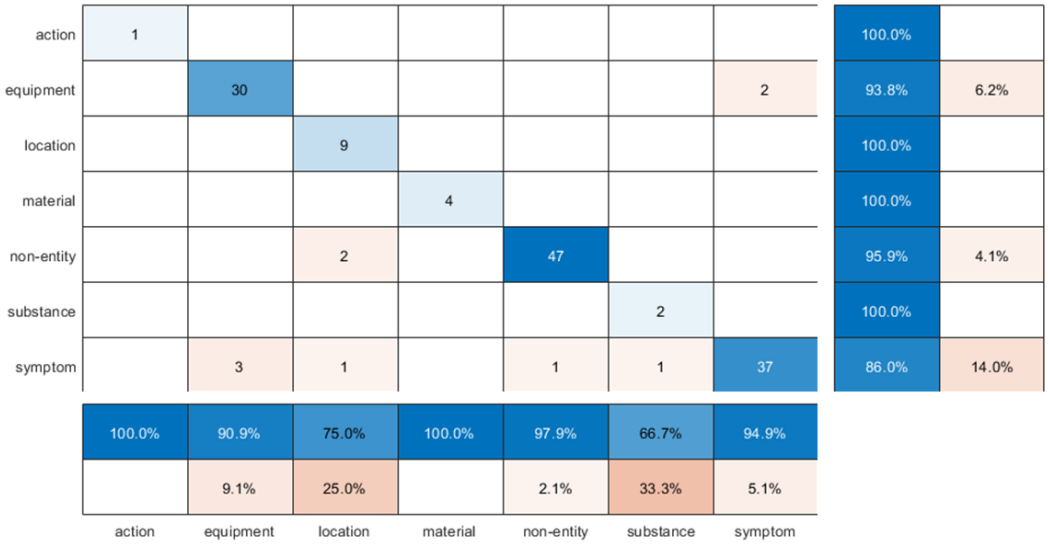

Fig. 7. Confusion chart of NER prediction for oversampled minority classes

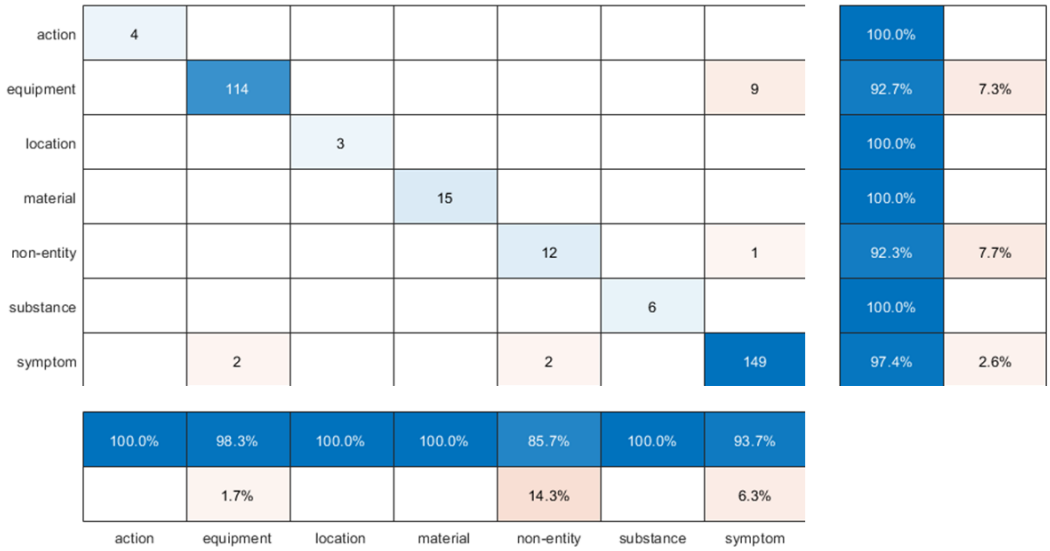

Fig. 8. Confusion chart of NER prediction with text preprocessing

Oversampling is a term used in manufacturing to describe a procedure where the output rate is purposefully boosted above the anticipated demand or regular capability. This may be done for several applications, including addressing production inefficiencies or managing unanticipated demand increases. Also, in equipment maintenance, to accommodate for downtime brought on by planned or unforeseen equipment repair, oversampling might be utilised. Manufacturers can make up for the loss of production capacity and keep up a steady supply to clients by producing more than the immediate demand.

On the other hand, the application of text preprocessing used in manufacturing, including technical documentation, customer feedback, and product descriptions, frequently includes noise or unimportant information. Text cleaning entails deleting or lessening noise, such as special letters, punctuation, and non-alphanumeric symbols, as well as removing HTML elements, URLs, or other patterns that are not significant to the analysis. Next, the HMM will be tested with new incoming factory reports from the personnels.

3.4 Test HMM

With accurate prediction of NER, the trained HMM can produce a series of precise entity for every word in the new factory reports within the continuous landscape of production line in manufacturing sector.

Figure 9 shows the new input of factory reports generated as the production is continuously operated. This information can be obtained from technician, operator, or engineer. The new input text is applied to test and validate the capability of the trained HMM with text preprocessing to recognizes the defined entity as shown in Figure 10. From the figure, it can be observed that each tokenized word has been correctly classified into the appropriate entity classes that were defined earlier. The improved precision in NER predictions has caused each word more meaningful within the scope of the specific domain entity compared to the general domain entity. Hence, NER precision has significantly eased decision-making and planning for maintenance diagnoses by technicians and engineers, thereby reducing the downtime of machine tools in the factory. Proper maintenance is vital for continuous manufacturing processes, and with improved NER precision, potential issues can be resolved proactively, ensuring optimal machine functionality.

```
str = [...
    "Coolant is pooling underneath sorter."
    "Sorter blows fuses at start up."
    "There are some very loud rattling sounds from the assembler."
    "programming crashed"];
```

Fig. 9. New input text from the maintenance personnel

Token	DocumentNumber	SentenceNumber	LineNumber	Type	Language	PartOfSpeech	Entity
"coolant"	1	1	1	letters	en	noun	substance
"pool"	1	1	1	letters	en	verb	symptom
"underneath"	1	1	1	letters	en	adverb	location
"sorter"	1	1	1	letters	en	noun	equipment
"sorter"	2	1	1	letters	en	adverb	equipment
"blow"	2	1	1	letters	en	verb	equipment
"fuse"	2	1	1	letters	en	noun	equipment
"start"	2	1	1	letters	en	noun	symptom
"loud"	3	1	1	letters	en	adjective	symptom
"rattling"	3	1	1	letters	en	noun	symptom
"sound"	3	1	1	letters	en	noun	symptom
"assembler"	3	1	1	letters	en	noun	equipment
"programming"	4	1	1	letters	en	noun	equipment
"crash"	4	1	1	letters	en	verb	symptom

Fig. 10. NER for the new input text

4. Conclusions

Unlocking the potential of diverse data formats, when meticulously analyzed, can avert unexpected, expensive, and time-consuming downtime, profoundly influencing manufacturing processes where a transformation can be achieved through precise maintenance data application and manipulation. Overall, training HMM for NER with oversampled preprocessed text outperformed the one without oversampling and text preprocessing due to consideration of significant minority classes' weight and words among entity class of 480 reports. This study has proved that by having either oversampling or text preprocessing technique, performance of NER prediction can be significantly improved up to 27.59%. With such prediction, manufacturers can predict equipment failures, provide a better plan maintenance task, and prevent expensive downtime by analysing data from sensors and other sources. For the application in manufacturing, oversampling can be implemented to manage unexpected increases of demand while text preprocessing can be employed to reduce noise that may jeopardize the analysis, coming from special characters, punctuation, and non-alphanumeric symbols, as well as removing HTML elements, URLs, or other particular patterns. In summary, the improvement in manufacturing prediction can result in more efficient use of resources, lower production costs, and faster time-to-market. In the future, the research is going to embark on the development and implementation of knowledge graph that can be integrated with the predictive maintenance information.

Acknowledgement

This work was supported by the Universiti Teknologi Malaysia under UTM Fundamental Research (Q.J130000.3823.22H48).

References

[1] Şeker, Gökhan Akın, and Gülşen Eryiğit. "Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content 1." *Semantic Web* 8, no. 5 (2017): 625-642. https://doi.org/10.3233/SW-170253

[2] Alfred, Rayner, Leow Ching Leong, Chin Kim On, Patricia Anthony, Tan Soo Fun, Mohd Norhisham Bin Razali, and Mohd Hanafi Ahmad Hijazi. "A rule-based named-entity recognition for malay articles." In *Advanced Data Mining and Applications: 9th International Conference, ADMA 2013*, Hangzhou, China, December 14-16, 2013,

Proceedings, Part I 9, pp. 288-299. Springer Berlin Heidelberg, 2013. https://doi.org/10.1007/978-3-642-53914-5_25

[3] Alfred, Rayner, Leow Chin Leong, Chin Kim On, and Patricia Anthony. "Malay Named Entity Recognition Based on Rule-Based Approach." *International Journal of Machine Learning and Computing* 4, no. 3 (2014). https://doi.org/10.7763/IJMLC.2014.V4.428

[4] Ibrahim, Musa Alhaji, Auwalu Yusuf Gidado, Abdulrahman Shuaibu Ahmad, Saidu Bello Abubakar, and Huang Kai. "Modelling and Virtual Manufacturing of a Flange Tube Using CAD/CAM Tools." *Journal of Advanced Research in Applied Mechanics* 43, no. 1 (2018): 1-7.

[5] Nazeer, KA Abdul. "Part-of-speech tagging and named entity recognition using improved hidden markov model and bloom filter." In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 1072-1077. IEEE, 2018.

[6] Baigang, Mi, and Fan Yi. "A review: development of named entity recognition (NER) technology for aeronautical information intelligence." *Artificial Intelligence Review* 56, no. 2 (2023): 1515-1542. https://doi.org/10.1007/s10462-022-10197-2

[7] Braunschweig, Katrin, Maik Thiele, Julian Eberius, and Wolfgang Lehner. "Enhancing named entity extraction by effectively incorporating the crowd." *Datenbanksysteme für Business, Technologie und Web (BTW) 2013-Workshopband* (2013).

[8] Chopra, Deepti, Nisheeth Joshi, and Iti Mathur. "Named entity recognition in Hindi using hidden Markov model." In *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, pp. 581-586. IEEE, 2016. https://doi.org/10.1109/CICT.2016.121

[9] Das, Arjun, and Utpal Garain. "CR-based named entity recognition@ICON 2013." *arXiv preprint arXiv:1409.8008* (2014).

[10] MathWorks. "FactoryReports Data Set." *The MathWorks, Inc.* 2020. https://www.mathworks.com/help/textanalytics/ug/data-sets-for-text-analytics.html.

[11] Hoque, A. S. M., P. K. Halder, M. S. Parvez, and Tamas Szecsi. "Integrated manufacturing features and Design-for-manufacture guidelines for reducing product cost under CAD/CAM environment." *Computers & Industrial Engineering* 66, no. 4 (2013): 988-1003. https://doi.org/10.1016/j.cie.2013.08.016

[12] Iglesias, Eva Lorenzo, A. Seara Vieira, and Lourdes Borrajo. "An HMM-based over-sampling technique to improve text classification." *Expert Systems with Applications* 40, no. 18 (2013): 7184-7192. https://doi.org/10.1016/j.eswa.2013.07.036

[13] Kim, Jae Kwon, Kyu Cheol Cho, Jong Sik Lee, and Young Shin Han. "Feature selection techniques for improving rare class classification in semiconductor manufacturing process." In *Big Data Technologies and Applications: 7th International Conference*, BDTA 2016, Seoul, South Korea, November 17-18, 2016, Proceedings 7, pp. 40-47. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-58967-1_5

[14] Kumar, Aman, and Binil Starly. ""FabNER": information extraction from manufacturing process science domain literature using named entity recognition." *Journal of Intelligent Manufacturing* 33, no. 8 (2022): 2393-2407. https://doi.org/10.1007/s10845-021-01807-x

[15] Li, Fulin, Yuanbin Song, and Yongwei Shan. "Joint extraction of multiple relations and entities from building code clauses." *Applied Sciences* 10, no. 20 (2020): 7103. https://doi.org/10.3390/app10207103

[16] Drovo, Mah Dian, Moithri Chowdhury, Saiful Islam Uday, and Amit Kumar Das. "Named entity recognition in bengali text using merged hidden markov model and rule base approach." In *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, pp. 1-5. IEEE, 2019. https://doi.org/10.1109/ICSCC.2019.8843661

[17] Shrivastava, Manu, Kota Seri, and Hiroaki Wagatsuma. "A Named Entity Recognition Model for Manufacturing Process Based on the BERT Language Model Scheme." In *International Conference on Social Robotics*, pp. 576-587. Cham: Springer Nature Switzerland, 2022. https://doi.org/10.1007/978-3-031-24667-8_50

[18] Mo, Hsu Myat, Khin Thandar Nwet, and Khin Mar Soe. "CRF-based named entity recognition for Myanmar language." In *Genetic and Evolutionary Computing: Proceedings of the Tenth International Conference on Genetic and Evolutionary Computing*, November 7-9, 2016 Fuzhou City, Fujian Province, China 10, pp. 204-211. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-48490-7_24

[19] Pasa, Luca, Alberto Testolin, and Alessandro Sperduti. "Neural Networks for Sequential Data: a Pre-training Approach based on Hidden Markov Models." *Neurocomputing* 169 (2015): 323-333. https://doi.org/10.1016/j.neucom.2014.11.081

[20] Pham, Minh Quang Nhat. "A Feature-Based Model for Nested Named-Entity Recognition at VLSP-2018 NER Evaluation Campaign." *Journal of Computer Science and Cybernetics* 34, no. 4 (2018): 311-321. https://doi.org/10.15625/1813-9663/34/4/13163

[21] Syachrul M. A. K., R. M., Moch Arif Bijaksana, and Arief Fatchul Huda. "Person entity recognition for the Indonesian Qur'an translation with the approach hidden Markov model-viterbi." *Procedia Computer Science* 157 (2019): 214-220. https://doi.org/10.1016/j.procs.2019.08.160

[22] Dixit, Sharad, Varish Mulwad, and Abhinav Saxena. "Extracting Semantics from Maintenance Records." *arXiv preprint arXiv:2108.05454* (2021).

[23] Balakrishnan, Vimala, and Ethel Lloyd-Yemoh. "Stemming and Lemmatization: A Comparison of Retrieval Performances." *Lecture Notes on Software Engineering* 2, no. 3 (2014). https://doi.org/10.7763/LNSE.2014.V2.134

[24] Wang, Jiahai, Wenming Yin, and Jianxiang Gao. "Cases Integration System for Fault Diagnosis of CNC Machine Tools Based on Knowledge Graph." *Academic Journal of Science and Technology* 5, no. 1 (2023): 273-281. https://doi.org/10.54097/ajst.v5i1.5664

[25] Wichmann, Pascal, Alexandra Brintrup, Simon Baker, Philip Woodall, and Duncan McFarlane. "Extracting supply chain maps from news articles using deep neural networks." *International Journal of Production Research* 58, no. 17 (2020): 5320-5336. https://doi.org/10.1080/00207543.2020.1720925

[26] Sari, Widia Permata, Moch Arif Bijaksana, and Arief Fatchul Huda. "Indexing name in hadith translation using hidden markov model (hmm)." In *2019 7th International Conference on Information and Communication Technology (ICoICT)*, pp. 1-5. IEEE, 2019. https://doi.org/10.1109/ICoICT.2019.8835296

[27] Wu, Lang-Tao, Jia-Rui Lin, Shuo Leng, Jiu-Lin Li, and Zhen-Zhong Hu. "Rule-based information extraction for mechanical-electrical-plumbing-specific semantic web." *Automation in Construction* 135 (2022): 104108. https://doi.org/10.1016/j.autcon.2021.104108

[28] Yin, Didi, Siyuan Cheng, Boxu Pan, Yuanyuan Qiao, Wei Zhao, and Dongyu Wang. "Chinese named entity recognition based on knowledge based question answering system." *Applied Sciences* 12, no. 11 (2022): 5373. https://doi.org/10.3390/app12115373

[29] Luo, Yin, Haishan Feng, Xuanlong Weng, Ke Huang, and Huang Zheng. "A novel oversampling method based on SeqGAN for imbalanced text classification." In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2891-2894. IEEE, 2019. https://doi.org/10.1109/BigData47090.2019.9006138