SEMI-SUPERVISED TOPIC MODELING FOR SHORT TECHNICAL TEXT

AN APPLICATION TO MANUFACTURING EXECUTION SYSTEM (MES) EXCEPTION DATA

CHELS JONES

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2155125

COMMITTEE

prof. dr. Eric Postma
ir. Bettina Soós

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

29-11-2025

WORD COUNT

8726

ACKNOWLEDGMENTS

I would like to thank my academic advisor, Dr. Eric Postma, and my internship advisor, Ronald den Hoed, for their guidance during the writing of my thesis. I would also like to thank Amgen Breda for providing this opportunity to apply what I've learned in my studies to a real-world and data-driven business case. Additionally, I am extremely grateful to my partner and my parents, as without their love and support this thesis and my studies would not have been possible.

SEMI-SUPERVISED TOPIC MODELING FOR SHORT TECHNICAL TEXT

CHELS JONES

Abstract

This thesis examines whether semi-supervised topic modeling via automatic label propagation can reliably classify manufacturing exception records in pharmaceutical manufacturing. Using Manufacturing Execution System (MES) exception records from Amgen Breda, where operators provide free-text descriptions containing root-cause information each exception, the study investigates if a contextual representation topic model can effectively propagate labels from a small labeled subset to a larger unlabeled subset, thereby reducing the need for large-scale manual labeling efforts. It then aims to train effective classifiers from the propagated dataset. Using automatic label propagation for classifying MES exception records in pharmaceutical manufacturing offers benefits for patient and worker safety, as well as material efficiency. Inspired by Babikov et al. (2023), Gottumukkala et al. (2025), and Kazanci (2025), a workflow using BERTopic (Grootendorst, 2022) was developed to cluster embeddings and propagate labels from a 20% labeled subset. Tuning improved propagation quality, increasing micro F1 to 0.54 and macro F1 to 0.60 on a held-out test set. Three training data variants were created from the propagated dataset to train Random Forest (RF) and Logistic Regression (LR) classifiers. Automatic label propagation improved cross-validation performance of the classifiers, achieving macro F1 scores of 0.79 (RF) and 0.65 (LR). On the test set, the best models achieved macro F1 scores of 0.56 (RF) and 0.53 (LR), with micro F1 around 0.45. High AUC values indicate meaningful class separation despite noise, with RF generally performing best. The findings show that BERTopic-based label propagation enhances classifier training in low-label settings, while refined label definitions and selective human review are needed for improved classifier performance on large, heterogeneous classes.

DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

Data Source: Amgen is the sole owner of the data used in this thesis and consents to use these data for the purpose of this thesis, provided that the confidentiality of its staff and of proprietary attributes in the data are kept confidential. A NUFFIC agreement to this effect was signed with Amgen before beginning the work on this thesis. All data was collected from their automated Manufacturing Execution System (MES), PAS X, and stored in their data lake. Data was extracted from this system using Amgen's SQL database within Databricks. No additional data was collected from operators or other staff. All personal, confidential and company proprietary information was removed from the data and replaced with semantically similar placeholders. Operators and quality assurance staff were only involved to help in labeling some data points as part of a semi-supervised model pipeline. A copy of the initial sample and the labeled sampel are included with this submission. An explanation of the datasets included with the thesis can be found in Appendix A.

Figures: All figures and tables were originally created by the author.

Code: Databricks 15.4.x-gpu-ml-scala2.12 was used as the environment in which all code was written and all models were run and tested. Open AI's Chat GPT (GPT-5) model was used for debugging code. All code was written in Python, version 3.11.11. The code files are attached to the submission of this thesis as .ipynb files. An explanation of the different code files and the parts of the thesis they relate to can be found in Appendix B. The following is a complete list of all Python libraries and packages used and their versions:

- bertopic: 0.16.1
- hdbscan: 0.8.38.post1
- imblearn: 0.11.0
- matplotlib: 3.7.2
- numpy: 1.23.5
- openai: 1.35.3
- openpyxl: 3.1.5
- pandas: 1.5.3
- requests: 2.31.0
- scipy: 1.11.1
- seaborn: 0.12.2
- sklearn: 1.4.2
- spacy: 3.7.2
- spellchecker (Symspell): 6.7.3
- umap: 0.5.6

Technology Use: Excluding the technologies already listed in the "Code" section above, the following technologies have been used in the preparation and writing of this thesis:

- *Spelling and grammar checking:* Overleaf's built-in spelling and grammar check tool
- *Layout, typesetting, and formatting overall:* Overleaf's LaTex editor
- *Reference management and citation formatting:* Zotero
- *Reference search:* Google Scholar, Arxiv
- *Formatting of tables:* Both Mistral's LLM Le Chat (Mistral-large-2407) and Open AI's LLM Chat GPT (GPT-5) were used to format tables in LaTeX code.
- *Creating figures:* Microsoft Powerpoint

1 PROBLEM STATEMENT AND RESEARCH GOAL

Manufacturing environments often generate large amounts of data related to machine diagnostics, procedural steps, and errors that occur. Many manufacturing environments rely on a Manufacturing Execution System (MES) to record any deviations from the standard operating procedure, also referred to as *exceptions*. At Amgen Breda, part of the US-based pharmaceutical manufacturing company Amgen, *exceptions* are documented with free-text descriptions provided by operators. Maintaining exception records for manufacturing procedures is required to meet regulatory requirements of the pharmaceutical industry. Exceptions also offer potential insights related to root cause analysis and continuous improvement. Although exceptions are already classified by the broad type of event that led to them, user descriptions entered by operators give more fine-grained details on what the process deviation was and how it was resolved, such as specific production line equipment affected, specific products involved, and sequences of events and actions that led up to and followed the exception.

For example, an exception may be assigned the event type "Tolerance violation" by the system, yet in the user description the operator may describe the exception as:

"During production Operator[NAME] noticed that 1 PFP was misplaced on conveyor plate, informed QA [NAME], not broken or damaged, scrapped as other and resumed production. Followed 10 cycles, no problems observed." This indicates that material was misplaced, which is likely due to an error in the task execution leading up to the event. This gives more valuable information about the root cause than the event type "Tolerance violation" alone.

These user descriptions provide a wealth of information for investigating common causes of exceptions. However, previous manual labeling efforts at Amgen have been time-consuming, costly and error prone due to a lack of automation and uniformity. Machine learning methods can automate the identification and classification of root causes within exception text data and reduce both time and human error in the classification process.

2 RESEARCH STRATEGY

The current literature on topic modeling and textual data mining related to manufacturing focuses mainly on academic studies involving large-scale literature reviews and meta-analyses of research themes in manufacturing (Sabbagh and Ameri, 2019, Xiong et al., 2019, J. Wang and Hsu, 2020, Sala et al., 2025, Xu et al., 2025). Few examples can be found in the literature that apply topic modeling and text mining techniques to real-

world manufacturing data. Despite this, the literature recognizes the need for research that bridges the gap between academia and the private sector in terms of the applicability of topic modeling and other Natural Language Processing (NLP) techniques (Sabbagh and Ameri, 2019, Sala et al., 2025, Xu et al., 2025). Topic modeling and textual data mining typically rely on one of three forms of modeling: unsupervised, supervised, and semi-supervised. Unsupervised topic modeling refers to methods where prior categorization or grouping of texts is not available (Vayansky and Kumar, 2020, Churchill and Singh, 2022, Rüdiger et al., 2022, Abdelrazek et al., 2023, Hankar et al., 2025).

Semi-supervised topic modeling brings many of the same benefits of supervised topic modeling in terms of drawing on a previous grouping system, while reducing the amount of labels and human supervision needed for classification (Mekala and Shang, 2020, Bu et al., 2023, Babikov et al., 2023, Kim and Lee, 2024, Ravenda et al., 2025, Gottumukkala et al., 2025, Kazanci, 2025). Label propagation through network structures, such as graph networks or clustering, is one such useful semi-supervised method that can use a small set of labeled data and apply the labels to unlabeled data based on the similarity of the data points (D. Zhou et al., 2004, Zhang et al., 2017, Y. Zhou et al., 2019, Bu et al., 2023, Babikov et al., 2023). Semi-supervised methods that use clustering of contextual representations of text data, such as embeddings, offer a way to propagate a small set of labels to a broader dataset in a weak supervision setting by semantic similarity, reducing the workload in terms of manual categorization and improving the accuracy of text classification (Rüdiger et al., 2022, Abdelrazek et al., 2023, Babikov et al., 2023, Bu et al., 2023, Hankar et al., 2025, Kazanci, 2025). Despite this, semi-supervised topic modeling, and label propagation specifically, are both comparatively underrepresented in the literature.

2.1 *Societal Relevance*

The literature on topic modeling demonstrates the need for methods that are resource and time efficient in terms of labeling data for classification (D. Zhou et al., 2004, Mekala and Shang, 2020, Babikov et al., 2023, Bu et al., 2023, Gottumukkala et al., 2025, Kazanci, 2025). Improving the efficiency and consistency of categorizing and exploring MES exceptions at Amgen Breda contributes to a safer pharmaceutical manufacturing and production environment in terms of faster issue resolution, lower waste, and greater insight into the root causes of production line exceptions, thus supporting both patient safety and environmental sustainability goals. Using semi-supervised label propagation with a small labeled set can also lead to better use of data in downstream tasks, such as training classifiers

to label new data points or generating large amounts of labels for use in database management and data retrieval systems. This makes applying machine learning techniques to the free-text user descriptions attached to MES exceptions an innovative and relatively underexplored application. It can contribute not only to covering gaps in the academic literature but also to helping bridge the gap between academic machine learning and its application to improve industrial reliability.

2.2 *Research Questions*

The goal of this thesis is to evaluate the potential of semi-supervised topic models based on transformer-based embeddings for improved categorization of technical and manufacturing-related text data with low amounts of human intervention. This thesis largely follows the methodology of Babikov et al. (2023) by using the contextual representation topic model BERTopic (Grootendorst, 2022) to cluster embeddings of MES exceptions and automatically propagate labels from a small labeled set to a broader unlabeled dataset via semantic similarity. Variations of the propagated dataset are then used to train two classifiers, Random Forest (RF) and Logistic Regression (LR). The strengths and weaknesses of the propagation and classifier training strategies are then compared and the resulting efficacy of the approach for classifying MES exception records is discussed in terms of its accuracy and robustness.

The following main research questions (RQ) and associated sub-questions (SQ) will be answered.

RQ 1: Can BERTopic accurately propagate labels for MES exceptions based on semantic similarity using transformer-based embeddings?

- ***SQ 1.1***: Does varying the training data by the confidence of the propagated labels included substantially affect the performance of downstream classifiers?

RQ 2: Can effective text classifiers be trained for MES exceptions using semi-supervision through automatic label propagation?

- ***SQ 2.1:*** What is the minimum supervision required, in terms of the amount of labeled data, to build an effective classifier for MES exceptions?
- ***SQ 2.2:*** Which of the downstream classifiers (RF or LR) performs better on MES exception text data with weak supervision?

2.3 *Findings*

The results of this research strategy showed that automatic label propagation markedly improved the training signal available to downstream classifiers. BERTopic label propagation achieved a macro F1 score of 0.60 evaluated on a held-out test set, using 20% labeled data in the training set. The training and tuning of downstream classifiers achieved cross-validation macro F1 scores of 0.65 - 0.72 and cross-validation micro F1 scores of 0.73 - 0.81. When evaluated on the held out test, the classifiers achieved macro F1 scores of 0.53 - 0.56 and micro F1 scores of 0.45 each. Therefore, although moderate propagation accuracy was achieved and the downstream classifiers performed well in cross-validation, they generalized less well on unseen data. However, error analysis through receiver operating curves and model classification reports revealed that classifiers were better at understanding which classes were more likely than at assigning the correct final label, particularly for larger classes. These results did not surpass the state of the art achieved by Babikov et al. (2023), who achieved a micro F1 score of 0.65 using DBSCAN clustering for automatic label propagation together with an SBERT neural classifier. However, given that Babikov et al. (2023) had access to a higher ratio of labeled data and trained a more advanced, but less interpretable, neural classifier on their propagated data, these results still show moderate success given the study conditions and show that automatic label propagation can be an effective strategy for training classifiers while reducing labeling workload.

3 LITERATURE REVIEW

Topic modeling aims at extracting themes from texts based on the patterns of words within them. Many current topic models rely on converting words in a document to a numerical representation. These numerical representations are generally either sparse vectors representing patterns of word co-occurrences or, in more recent topic models, dense embeddings of semantic relationships (Röder et al., 2015, Rüdiger et al., 2022, Churchill and Singh, 2022, Abdelrazek et al., 2023). Topic models use numerical representations of text data to convert thematic patterns in the text to human-interpretable representations, usually in the form of a list of the most representative words for a topic (Vayansky and Kumar, 2020, Rüdiger et al., 2022, Abdelrazek et al., 2023). Topic modeling has most notably been applied in large research studies and systematic reviews, particularly in the social sciences, marketing research, bioinformatics and economics (, Xiong et al., 2019, Rüdiger et al., 2022, Egger and Yu, 2022, Abdelrazek et al., 2023, Mishra, 2024). The potential of topic modeling is broad and holds

promise for uncovering patterns in domains such as business, medicine, and engineering, and manufacturing (J. Wang and Hsu, 2020, Rezaei et al., 2024, Sala et al., 2025, Ma et al., 2025, Xu et al., 2025). The following literature review outlines the main types of topic models, the application of topic modeling to short, technical texts, an overview of semi-supervised topic modeling methods, and an exploration of one of the most recent and innovative topic models, BERTopic (Grootendorst, 2022).

3.1 *Topic modeling typology*

Several existing systematic reviews of topic modeling have proposed different frameworks to categorize topic models. Here, a combination of the typologies proposed by Rüdiger et al. (2022) and by Abdelrazek et al. (2023) are used to outline the different types of topic models. Topic models can generally be categorized in three ways: based on their approach to extracting meaning from words, the way in which they form topics and the degree to which they use any neural architecture. Figure 1 below presents the overlapping nature of the three groups of topic models.

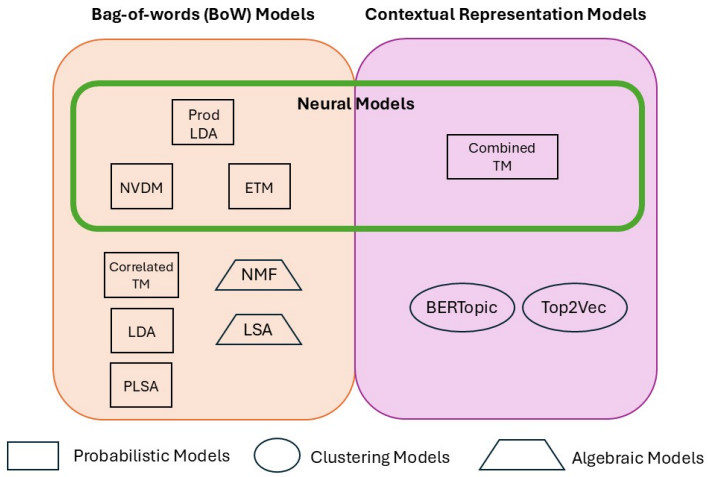

Figure 1: Model families of major topic models in literature (TM = topic model, LDA = Laten Dirichlet Allocation, PLSA = Probabilistic Latent Semantic Analysis, LSA = Latent Semantic Analysis, NMF = Non-negative Matrix Factorization, NVDM = Neural Variational Document Model, ETM = Topic Model in Embedding Space)

Topic models extract meaning from words and sentences based on either a Bag of Words (BoW) or a contextual representation approach. BoW topic models model words in a document as a document-term matrix that counts

the co-occurrences of words within a document, creating sparse vector representations of topics. Therefore, BoW models lack the ability to include semantic information such as word order, grammar or similarity to other words. Some examples of BoW models include Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Latent Semantic Analysis (LSA) (Dumais, 2004) and Correlated TM (Blei and Lafferty, 2007). Contextual representation models instead model words as dense semantic embeddings, series of non-zero numbers, often produced by transformer models. These dense semantic embeddings, unlike sparse vectors, are able to capture semantic information that BoW models lack. In reality, there is not always a clear line between these two broad groups. Some newer topic models, such as BERTopic (Grootendorst, 2022) or Combined TM (Bianchi et al., 2021), technically combine both BoW and contextual representation approaches to create topic representations.

Topic models also differ in how they form topics from words, generally using either an algebraic, probabilistic or clustering approach. The first topic models to emerge in the 1990s were algebraic models, such as LSA, Non-Negative Matrix Factorization (NMF) (Paatero and Tapper, 1994) and Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999), which use linear algebra and matrices to condense the dimensionality of the matrix and form topic representations. Probabilistic models form topics by constructing a probability distribution for each document over the words in the document. This has the advantage of allowing a single document to be labeled with multiple topics (i.e. 60% Topic A and 40% Topic B), similar to real-world conditions. LDA is the most well-known probabilistic model and the most frequently referenced topic modeling algorithm in the literature (
Rüdiger et al., 2022, Abdelrazek et al., 2023). Clustering topic models are most often also contextual representation models, as the clustering approach relies on using dense embeddings to group word representations in 3-D space, with each cluster forming a topic. Examples of these models include Top2Vec and BERTopic (Angelov, 2020, Grootendorst, 2022).

Topic models of all previous varieties can also make use of neural architecture. Examples include Combined TM (Bianchi et al., 2021) as well as Product of Experts LDA (ProdLDA) (Srivastava and Sutton, 2017), Neural Variational Document Model (NVDM) (Miao et al., 2016) and Topic Model in Embedding Spaces (ETM) (Dieng et al., 2019). Neural models use neural networks to learn topics directly from data instead of using the manual sampling procedures found in classical models. Neural topic models can use both BoW vectors and dense semantic embeddings as input.

3.2 *Influence of text length on topic model selection*

The domain significantly influences the choice of topic model, as text length affects performance. Topic modeling has been applied across diverse fields, including social sciences (Söderwall and Telešova, 2025), bioinformatics (Y. Wang et al., 2022), medicine (Ma et al., 2025), marketing (Mishra, 2024), computer science (Wankmüller, 2023, Huseynova and Isbarov, 2024, Rezaei et al., 2024), manufacturing (Sabbagh and Ameri, 2019, Xu et al., 2025, Sala et al., 2025), social media (Egger and Yu, 2022, Laureate et al., 2023) and news (Asas et al., 2025). Most studies focus on long, complex academic texts, while social media studies address shorter, simpler, and erratic content.

Research has shown that short text presents unique challenges for topic models, as many topic models rely on the word co-occurrence patterns within documents to infer the topic or mixture of topics present in that document. With short text, many word co-occurrence patterns either never occur or occur infrequently, known as sparsity. This makes inferring the topic based on word co-occurrences for a short document more difficult than for long text (Yan et al., 2013, Qiang et al., 2022, Egger and Yu, 2022, Laureate et al., 2023). This presents an interesting challenge for the present study, as MES exception messages are generally less than 1,000 characters long, while most topic modeling related to manufacturing has modeled topics within the scientific and academic literature on manufacturing, primarily dealing with long and complex texts (Xiong et al., 2019, , Sabbagh and Ameri, 2019, J. Wang and Hsu, 2020,Sala et al., 2025, Xu et al., 2025).No studies were found that focus on using topic modeling to analyze topics in manufacturing similar to MES exception messages, such as quality assurance documentation or error messages. However, similar studies using contextual representation models to classify software descriptions and defects were found in the literature (Zhang et al., 2017, Bu et al., 2023, Gottumukkala et al., 2025). Problems with parsing sparseness in short texts occur more often when using probabilistic and BoW topic models, such as LDA or PLSA (Yan et al., 2013,Qiang et al., 2022). Incorporating pre-trained word embeddings has been proposed in the literature as one way to improve the use of topic models on short text, since using embeddings allows the model to recognize semantically related words to substitute in sparse co-occurrence patterns (C. Li et al., 2017, X. Li et al., 2019, Qiang et al., 2022).

3.3 *Unsupervised and semi-supervised learning in topic models*

Topic modeling in the literature primarily consists of unsupervised and semi-supervised methods. Most studies focus on unsupervised learning, where large text corpora are explored without known labels. This allows researchers to identify hidden themes within text (Rüdiger et al., 2022, Abdelrazek et al., 2023, Hankar et al., 2025). Unsupervised models are commonly evaluated using intrinsic validity metrics such as perplexity (Rüdiger et al., 2022, Hankar et al., 2025, Söderwall and Telešova, 2025), topic coherence (Newman et al., 2010, Röder et al., 2015, Hankar et al., 2025), topic diversity (Chang et al., 2009, Dieng et al., 2019), and human interpretability (Xiong et al., 2019, J. Wang and Hsu, 2020, Egger and Yu, 2022, Mishra, 2024, Sala et al., 2025, Ma et al., 2025). Unsupervised topic modeling is also frequently used to generate topic-derived labels for training downstream classifiers (Kazanci, 2025, Gottumukkala et al., 2025) by manually converting topic representations into concise class labels.

Semi-supervised topic modeling extends this idea by using a small set of labeled examples ("seeds") to guide topic formation. Unlike supervised learning, semi-supervised methods do not require full labeling of the dataset. Instead, they use the available labeled subset to steer topic clustering and propagate labels (D. Zhou et al., 2004). Contextual representation models are particularly effective for semi-supervised tasks due to their ability to capture semantic and contextual relationships, leading to more coherent clustering (Zhang et al., 2017, Babikov et al., 2023, Bu et al., 2023). These methods follow seeded-word or seeded-label strategies. Figure 2 below provides a schema of the different types of semi-supervised topic modeling.

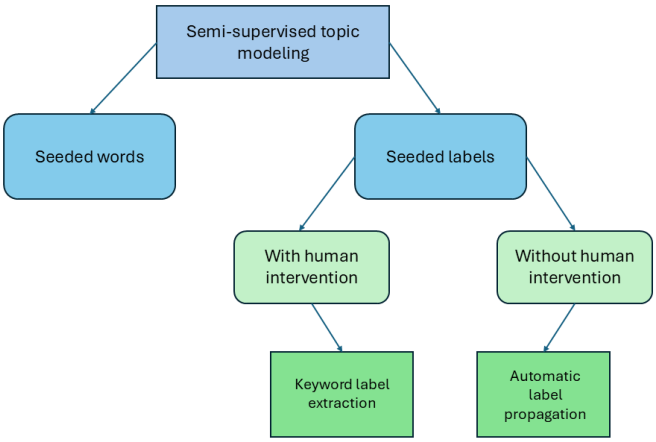

Figure 2: Types of semi-supervised topic modeling

Seeded-word approaches provide predefined terms representing concepts of interest, which act as anchors for clustering and push the model toward semantically related groupings (Mekala and Shang, 2020, Ravenda et al., 2025). Seeded-label approaches, also known as label propagation, begin with a subset of documents that already have class labels. Clustering organizes documents by semantic similarity, and labels are propagated according to two assumptions: "points near a labeled point likely share its label, and points within the same cluster likely share the same label" (D. Zhou et al., 2004). Majority voting is then used to assign labels to unlabeled documents (Babikov et al., 2023, Bu et al., 2023). This approach is resource-efficient because it expands a small labeled subset into a larger labeled dataset suitable for classifier training (Gottumukkala et al., 2025, Kazanci, 2025), metadata creation, or semantic retrieval tasks (Wankmüller, 2023, Rezaei et al., 2024).

Label propagation may involve human oversight or be fully automatic. Human-guided approaches review a sample of propagated labels before applying them to the full dataset, whereas automatic methods rely entirely on clustering structure or confidence scores. Babikov et al. (2023) compared both strategies and trained classifiers on each resulting dataset. Their automatic label propagation method, without human intervention, produced classifiers with micro F1 scores between 0.65 - 0.70, a substantial improvement over the 0.38 achieved by training on the original labeled subset. Human-guided propagation produced even stronger results, with micro F1 scores of 0.76 - 0.82.

3.4 *BERTopic for semi-supervised topic modeling*

Models based on embeddings, also called contextual representations, are the current state-of-the-art for unsupervised and semi-supervised topic modeling. Across the literature, BERTopic is the most widely applied contextual model, used for tasks ranging from exploratory topic discovery to label extraction and propagation. BERTopic (Grootendorst, 2022) provides an end-to-end pipeline that embeds text, reduces dimensionality with UMAP (McInnes et al., 2020), clusters with HDBSCAN (McInnes et al., 2017) and extracts keywords using CountVectorizer and class-based TF-IDF (c-TF-IDF). Figure 3 below further illustrates BERTopic's approach to topic modeling. In comparative studies, BERTopic has outperformed probabilistic models such as LDA and PLSA, and other contextual models like Doc2Vec and Top2Vec, particularly in topic coherence and ease of use (Egger and Yu, 2022; Gan et al., 2024; Xu et al., 2025).

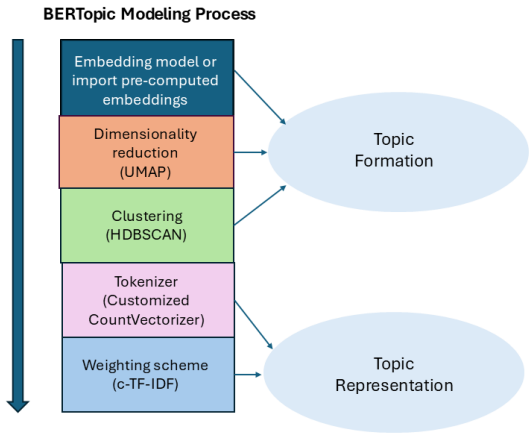

Figure 3: Modular modeling process of BERTopic

Although frequently used for unsupervised analysis, BERTopic also supports semi-supervised workflows. Gottumukkala et al. (2025) used BERTopic with human-guided labeling to cluster software defect descriptions, extract labels from topic keywords, and train downstream classifiers, achieving a 0.95 F1 score on their downstream classifier (Gottumukkala et al., 2025). Bu et al. (2023) also applied BERTopic for software subclassification, where keyword-derived labels improved their neural classifier from a 0.92 - 0.96 weighted F1 score (Bu et al., 2023). Kazanci (2025) similarly used human guidance with BERTopic-derived labels to train a call-center classifier with a 0.87 F1 score (Kazanci, 2025).

BERTopic can also propagate existing labels without keyword extraction (i.e. automatically) via its clustering mechanism. Babikov et al. (2023) compared two approaches: fully automatic label propagation using clustering and majority voting, and a human-guided method using expert-selected keywords. Both produced datasets for classifier training. Although human-guided labels yielded higher F1 scores from their trained neural classifier (0.76 - 0.82), the automatic approach still achieved 0.65 - 0.70 and substantially outperformed the neural classifier trained only on the original labeled subset (0.38) (Babikov et al., 2023).

3.5 *Synthesis*

Although topic modeling has been used to analyze academic research and innovation trends in manufacturing, its application to production line data remains a novel contribution to the literature. Applying topic modeling to Manufacturing Execution System (MES) data presents challenges due to the short length of MES exception messages (median length < 400 characters, see Figure 18 in Appendix A). Contextual representation models are at the forefront of topic modeling, offering advantages such as improved topic coherence and human interpretability over probabilistic and BoW models like LDA. Semi-supervised contextual representation models have already been successful in training classifiers for software categorization (Zhang et al., 2017, Bu et al., 2023) and in automatic label propagation (Babikov et al., 2023). This thesis applies the advances of semi-supervised automatic label propagation and contextual representation topic modeling to make a novel contribution to the literature by applying them to short, technical MES exceptions. The state-of-the-art found in the literature for DBSCAN clustering-based automatic label propagation combined with downstream classifier training was found in Babikov et al. (2023), achieving a micro F1 score of 0.65 with 25% labeled data in the training set. Following Babikov et al. (2023), this thesis uses BERTopic's UMAP dimensionality reduction and HDBSCAN clustering components in a semi-supervised approach to propagate labels using 20% labeled data in the training set, labeled by domain experts, to a broader dataset. After evaluating propagation accuracy, the expanded labeled dataset is used to train classifiers for categorizing new MES exceptions.

4 METHODOLOGY

Following Babikov et al. (2023), this thesis applied BERTopic, including UMAP and HDBSCAN, to cluster exception embeddings and perform semi-supervised automatic label propagation. Different version of the

propagated dataset were then used to train Logistic Regression (LR) (Berkson, 1944) and Random Forest (RF) (Breiman, 2001) models as downstream classifiers. RF and LR classifiers were chosen as they were shown in several studies to work well for unbalanced datasets with linear and non-linear relations between features (Bu et al., 2023, Gottumukkala et al., 2025, Kazanci, 2025). Linear models were also found to be the most easily interpretable, while ensemble models often achieved higher accuracy (Gottumukkala et al., 2025, Kazanci, 2025). Other studies used neural network models for classification based on propagated data (Mekala and Shang, 2020, Babikov et al., 2023, Bu et al., 2023). Although these models may offer slight improvements in accuracy, their complexity of hyperparameter tuning and their lack of interpretability make them poor choices for our context.

Both macro and micro F1 scores are used as evaluation metrics in the methodology, with macro F1 used to measure the accuracy of label propagation and both micro and macro F1 used to measure the accuracy of the classifiers trained on the propagated data. Although the most common variation of F1 used in the surveyed literature is the micro F1 score (Babikov et al., 2023, Gottumukkala et al., 2025, Kazanci, 2025), macro F1 is included to account for large differences in class sizes in the dataset. While the micro F1 score weighs the individual F1 scores for each class according to their size, the macro F1 score weighs each class' individual F1 score equally. Including both as evaluation metrics helps to give a better overall picture of how each model performs on both smaller and larger classes.

Before turning to a detailed description of the methodology, a high-level overview is provided. Figures 4 and 5 together illustrate the full methodological process.

The basis of the study are the free-text user descriptions attached to the MES exceptions. These contain text of multiple sentences, up to 1,000 characters long in total (see Figure 17 in Appendix A). An example of these user descriptions and their content, after cleaning and removing confidential information, can be found in Table 8 in Appendix A.

First, exceptions with user descriptions that are too short to retain relevant information were removed. Once these exceptions were removed, an initial random sample of 9,989 exceptions, stratified by event type, was taken as a basis for further cleaning and preprocessing. Extensive text cleaning was performed for the exceptions in this sample. For each user description in the sample, the company's proprietary information and staff names were removed, spelling correction was performed, and punctuation marks were standardized. Dense semantic embeddings of each cleaned user description were created using Open AI's text-embedding-large-3 model to create a vector search index within the company's Databricks environment (AI, 2025). These embeddings, of 3,072 dimensions each,

were then reduced using Principal Component Analysis (PCA) to preserve semantic information while reducing the computational load for the models.

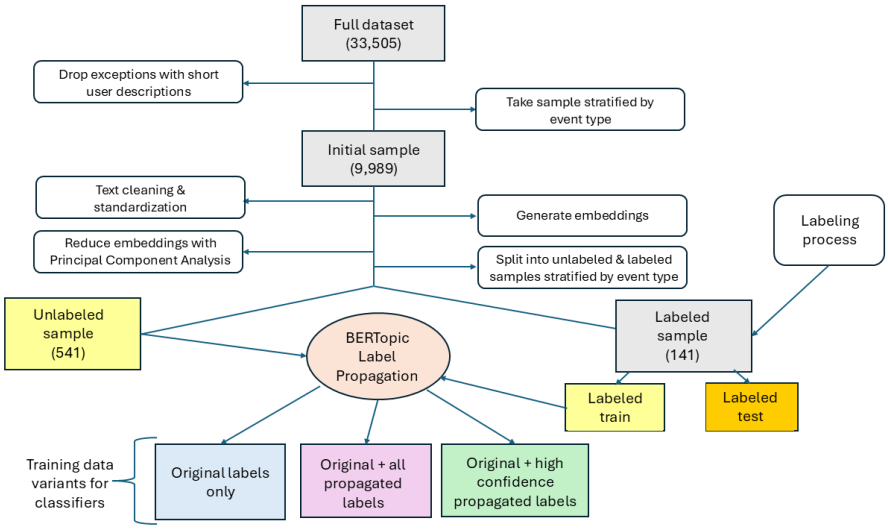

Figure 4: Methodological steps including data preparation, sampling, BERTopic label propagation and creating training data variants

The cleaned and processed initial sample was then divided into one unlabeled sample and one sample for labeling by domain experts in manufacturing and quality assurance departments. The results of the labeling process were then split into labeled train and labeled test data. The labeled training data was appended to the unlabeled data and used to train a BERTopic model to propagate labels to unlabeled data through HDBSCAN clustering. The accuracy of the propagation was evaluated on the held-out labeled test set using the macro F1 score. After training and evaluating the initial BERTopic model, hyperparameter tuning was performed to improve the propagation accuracy.

Once a sufficiently accurate propagation process was completed using BERTopic, the propagated dataset was filtered into three data set variants: a dataset that includes only original labels from the labeled training data, one that includes both original and all propagated labels, and one that includes all original labels and only those propagated labels with a clustering membership strength greater than 0.6. Both an RF and an LR multiclass classifier were trained using the three dataset variations and tuned through stratified K-fold cross validation (CV). The final best model for each classifier / dataset combination was then evaluated on the held-out labeled test set using both macro and micro F1 scores. Figure 5 illustrates the classifier training process.

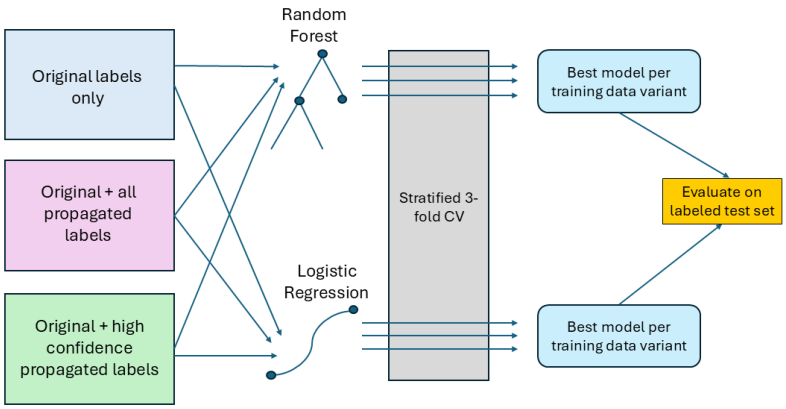

Figure 5: Process for using training data variants to train and evaluate classifiers

The following sections outline the dataset used, the data cleaning and preprocessing undertaken, the set-up and hyperparameter tuning of the BERTopic label propagation, and the training and tuning of the multi-class classifiers on the label-propagated data.

4.1 *Experimental set-up*

All experiments were run within the company's Databricks environment using Python. A constant random state variable was initiated for any stochastic processes throughout the environment. Sci-kit learn (Pedregosa et al., 2011) was used extensively for feature extraction, hyperparameter tuning, running classifiers, and evaluation.

4.1.1 *Dataset*

The initial dataset comprised 33,505 MES exceptions with 16 attributes, extracted from the company's MES software on 22-10-2025, covering exceptions from October 2022 to October 2025. Each row of the dataset relates to an exception registered in the MES, which is a deviation from the standard operating procedure that must be logged and explained for regulatory compliance. Each exception contains unique identifiers, such as the exception number and ID. Two of the most informative columns in the dataset are the event type and the user description. While the event type registers a general category regarding why the exception was raised, the user description is where operators must enter a free-text description of what specifically triggered the exception (e.g., mechanical failure, poor

product quality), who was involved, what product or procedure was affected, and how the exception was addressed, resolved, or escalated for further attention. Therefore, although the event types provide a general picture of the category of the exception, the user description field holds the most information regarding what exact errors or activities led to exceptions and how operators handled them. Table 1 below indicates each attribute in the dataset and what it represents. Figures 19 and 20 in Appendix A provide more details on the distribution of event types in the dataset and the characteristics of the exceptions.

Feature Name	**Description**
exceptionnumber	Unique identifier
exceptionid	Unique identifier
date	Date and time at which exception occurred
ebrcategory	String indicating the Electronic Batch Record (EBR) type
eventtype	Broad category for the exception generated by the system
userdescription	Free-text description of the actions taken to resolve or handle the exception, entered by the operator
clusternr	Unique identifier for a cluster (multiple messages that relate to the same exception procedure)

Table 1: Descriptions of attributes in the primary dataset

4.2 *Data preparation*

Additional columns were created for the cleaned user description (userdescclean) and for the calculated user description length (userdesclen). The first step in data preprocessing was to remove user descriptions that had little relevant textual information. User descriptions that were null or empty white space and user descriptions where the length of the user description was less than 11 characters were removed, as descriptions less than this length were determined in the initial analysis to only carry placeholders such as "—" or other non-informative message content. This resulted in the removal of approximately 1.7% of the total rows in the dataset. Figures 17 and 18 in Appendix A show that this removal did not alter the median length of user description messages and therefore is unlikely to significantly affect model results. To create a more compact basis for analysis, a random sample of a maximum of 10,000 rows was taken from the larger dataset after removing irrelevant data. This sample was stratified by event type, resulting in 9,989 rows.

Regex expressions were used extensively to clean and standardize the user description attribute. Names, roles, numbers, and alphanumeric strings were replaced with placeholders to preserve semantic relationships. This ensured staff privacy and protected proprietary information while also preventing the model from clustering based on irrelevant details, such as operator names or procedural codes. Special punctuation marks were removed or replaced, and frequently occurring phrasing was standardized for consistency. Stopwords and punctuation were left in the cleaning process, as they play an important role in accurately modeling sentence semantic relationships when converted to embeddings. Spelling correction was performed using the Symspell `spellchecker` Python package. Once the text was cleaned and standardized, spelling correction was performed using the Symspell Python package with a standard English correction library downloaded from Symspell's Github documentation (Garbe, 2025). A custom dictionary of commonly misspelled manufacturing terms found in the user descriptions was concatenated to the standard English dictionary to form the full dictionary used for spelling correction. A precise description of the placeholders used and the procedure followed can be found in Appendix A. Pseudocode for the data cleaning process can be found in Appendix B.

4.3 *Feature extraction*

The cleaned, processed and spelling corrected user descriptions from all exceptions were converted to semantic dense vector embeddings using Open AI's `text-embedding-3-large` model (AI, 2025). The cleaned embeddings were converted to a vector search index using the Open AI embedding model within the company's Databricks environment. This produced semantic dense vector embeddings of 3072 dimensions for each exception. Finally, the embeddings were reduced in dimensionality to retain 95% of the total variance of their features using Principal Component Analysis (PCA), via the `PCA` package from `Scikit-learn` (Pedregosa et al., 2011), with the constant random state of the environment. The choice to reduce the dimensionality of the embeddings was based on research showing that the use of PCA to further reduce the feature dimensions prior to UMAP generally did not result in loss of information while significantly reducing the computation time (McInnes et al., 2020). Each reduced embedding was attached to its corresponding exception.

4.4 *Labeled and unlabeled data samples*

The initial sample was further divided into a labeling sample and an unlabeled sample. For the labeling sample, a maximum of 150 rows were taken at random to be split into batches and labeled with categories by domain experts. Due to rounding and difficulty labeling a small number of exceptions, this resulted in 141 total exceptions in the labeled sample. An explanation of how the labeling process was carried out can be found in Appendix A. For the unlabeled sample, a maximum of 550 rows were taken at random and also stratified by event type, excluding those rows already included in the labeling sample, resulting in 541 total rows. This sample was left unlabeled to be used as a basis for label propagation.

The 141 labeled exceptions were randomly split into a training set and a test set using an 80/20 split, stratified by label class, ensuring that at least one instance of each class was present in both sets. Two classes with only one instance each were dropped to avoid issues during stratified K-fold CV and to prevent negative effects on the F1 score. This resulted in 108 labeled exceptions used as training data and 29 labeled exceptions used as a held-out test set, each with 9 classes. The labeled training data was appended to the unlabeled sample for label propagation, with the 108 labeled exceptions acting as partial supervision. The held-out labeled test set was used to evaluate the results of label propagation and trained classifiers. Table 2 below indicates the relative class counts and their imbalanced nature in the labeled train and labeled test datasets. Methods such as SMOTE were not used to rebalance the dataset, as traditional SMOTE algorithms do not interpolate well for high dimensional-data with very large feature spaces, such as embeddings (Blagus and Lusa, 2013, Y. Li et al., 2025).

Table 2: Distribution of assigned labels across train and test splits

Assigned Category	**Train Count**	**Train %**	**Test Count**	**Test %**
Entry error	17	15.74%	5	16.06%
Exception referral	4	3.70%	1	3.65%
Machine performance – Mechanical failure or jamming	20	18.52%	5	18.25%
Machine performance – Other	7	6.48%	2	6.57%
Machine performance – Packaging/labelling issue	17	15.74%	5	16.06%
Machine performance – Software / IT component failure	18	16.67%	5	16.79%
Material defect	10	9.26%	2	8.76%
Reconciliation – Material balance	3	2.78%	1	2.92%
Task execution error	12	11.11%	3	10.95%
Total	**108**	**100%**	**29**	**100%**

4.5 *BERTopic label propagation*

The basis of analysis for label propagation was the combination of the unlabeled sample (541 unique exceptions) and the labeled training data (108 unique exceptions), which form the propagation training data (649 unique exceptions). This resulted in a label-to-unlabeled ratio of roughly 1:5, or 20%, similar to that used in Babikov et al. (2023). A lower ratio for this study was chosen to test whether accurate propagation and classifiers could be trained using even less labeled data and to preserve a higher amount of total training data. An initial BERTopic model was run using the labeled training data as partial supervision to guide the HDBSCAN clustering. This initial model used the default parameters for the UMAP and HDBSCAN algorithms, shown in Table 10 in Appendix A. The labels were propagated to unlabeled data points within the same cluster based on majority voting. This process is depicted in Figure 6 below. The confidence of each label was determined by the membership strength output for each exception by the HDBSCAN algorithm. The results of the initial label propagation were evaluated on the held-out test set using both micro and macro F1. The state-of-the-art in literature for automatic label propagation, such as that performed by Babikov et al.(2023), was able to achieve a micro F1 score of approximately 0.65 using 25% labeled data and a DBSCAN clustering algorithm. Our methodology is expected to achieve a similar, yet slightly smaller, F1 score using the HDBSCAN algorithm with only 20% labeled data.

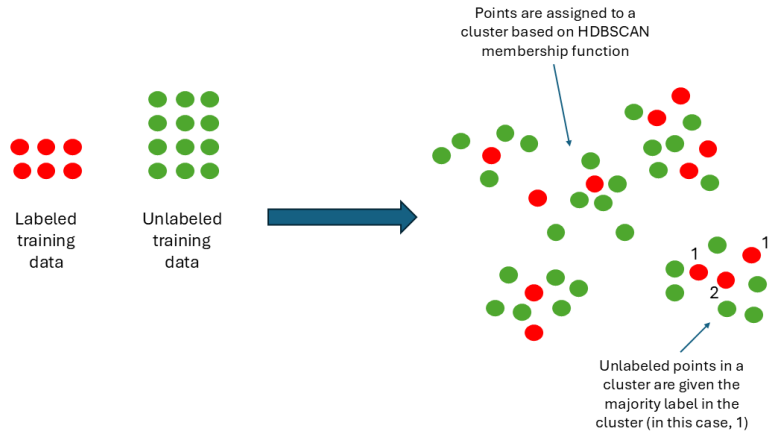

Figure 6: Process of label propagation from labeled to unlabeled data points

Hyperparameter tuning for the BERTopic propagation model was done in a three-step process. First, tuning was used to find the UMAP hyperparameters that produced the highest macro F1 score. The best UMAP hyperparameters were then fixed, and successive BERTopic models were run with a changing HDBSCAN hyperparameter grid. Finally, the best UMAP and HDBSCAN hyperparameters were fixed, and a final hyperparameter tuning was performed on the `nr_topic` parameter of BERTopic. The sets of hyperparameters used in the BERTopic tuning strategy can be found in Table 13 in Appendix A. The best performing parameters were used for a new round of label propagation, which was again evaluated on the held-out test set.

4.6 *Training classifiers*

RF and LR classifiers were selected to evaluate the impact of label propagation on predictive performance. These models were chosen to represent complementary paradigms: an ensemble of decision trees capable of capturing nonlinear relationships (RF) and a regularized linear classifier that provides interpretable decision boundaries (LR). Each classifier was trained and tuned using each of the following training data variants:

1. **Ground-truth dataset:** contained only the 108 labeled training rows combined with the unlabeled sample.
2. **Fully propagated dataset:** included all ground-truth labels together with all labels propagated by the semi-supervised BERTopic propagation model, regardless of confidence.
3. **High-confidence propagated dataset:** consisted of all ground-truth labels plus only propagated labels with clustering membership strength ≥ 0.6.

Stratified three-fold CV was used to optimize the hyperparameters of each classifier, with macro F1 as the evaluation metric. For comparability, each classifier used the same random sample of hyperparameter combinations across all three variants of the dataset. Each classifier and training dataset combo were put through hyperparameter tuning using three-fold stratified CV. The resulting best model for each classifier / training dataset combo was then evaluated once on the held-out test set based on both their macro and micro F1 score. Table 16 in Appendix A indicates the hyperparameter combinations tested as part of the hyperparameter tuning for each classifier. This resulted in the following total number of fits for each classifier.

- **RF:** 20 random hyperparameter combinations × 3-fold CV x 3 datasets = 180 fits
- **LR:** 20 random hyperparameter combinations × 3-fold CV x 3 datasets - 180 fits

Detailed information regarding the hyperparameter training architecture can be found in Appendix A.

5 RESULTS

5.1 *BERTopic label propagation*

Table 3 below shows the results of two BERTopic propagation models. The initial BERTopic propagation model showed low accuracy of the propagated labels compared to the original labels in the test set, but reached 100% % label coverage, with all unlabeled points receiving a label. The tuned BERTopic propagation model produced substantially higher accuracy of the propagated labels compared to the test set. By tuning the BERTopic propagation model, the micro F1 score of the propagated labels compared to the test set increased from 0.38 to 0.54 and the macro F1 increased from 0.51 to 0.6. The macro F1 score remained higher than the micro F1 score for both models. Although label coverage decreased slightly after tuning the BERTopic propagation model, the increase in both micro and macro F1 scores substantially outweighed the loss in total labels.

Table 3: Comparison of evaluation metrics between initial and tuned BERTopic propagation models

Metric	Initial propagation model	Tuned propagation model
Micro F1	0.38	0.54
Macro F1	0.51	0.60
Weighted F1	0.38	0.53
Label coverage	100%	82.8%

The confidence in the label assigned to an exception, proxied by the HDBSCAN membership strength of the data point within a cluster, also increased substantially between the initial and tuned propagation model. Figures 7 and 8 show the distribution of membership strength values for labeled exceptions after each propagation model.

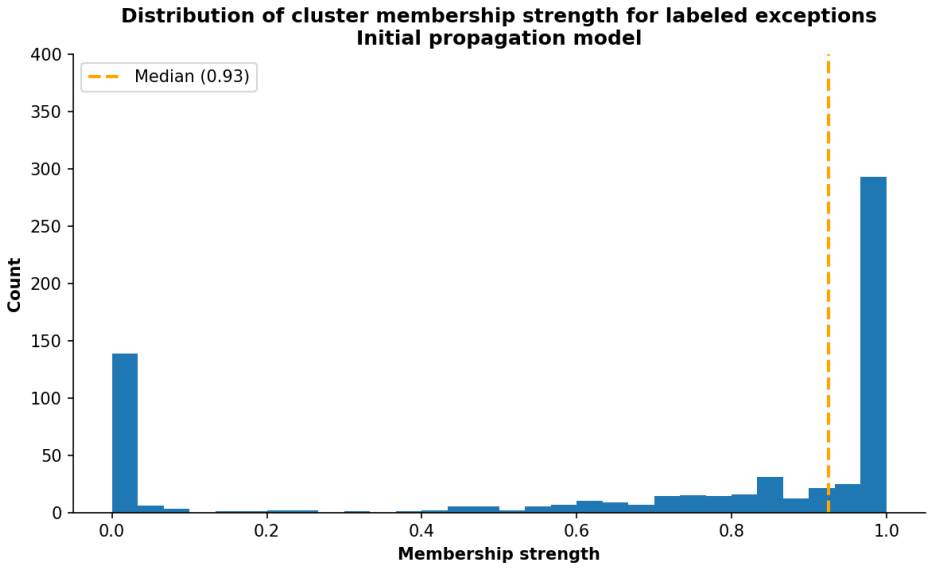

Figure 7: Distribution of label confidence according to cluster membership strength for the initial BERTopic propagation model

Figure 8 shows that the median membership strength for labels of the tuned propagation model reached 100%. This indicates an improved cluster structure in terms of homogeneity and separation moving from the initial to the tuned BERTopic propagation model.

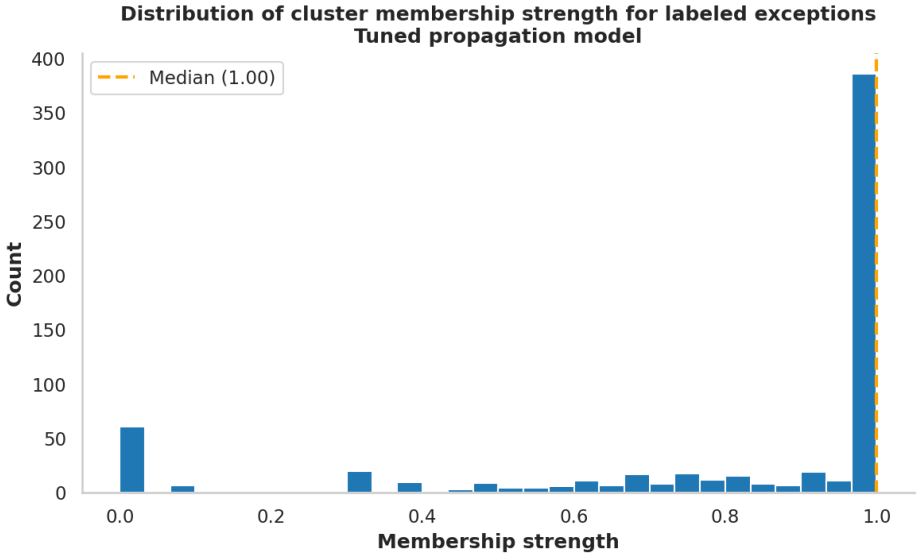

Figure 8: Distribution of label confidence according to cluster membership strength for the tuned BERTopic propagation model

The ratio of original to propagated labels within each class remained relatively stable between the initial and tuned propagation models. The same five classes remained the largest for both propagation models, although in different relative positions, as shown in figures 9 and 10, which show the number of labeled exceptions in each class and whether they have original or propagated labels.

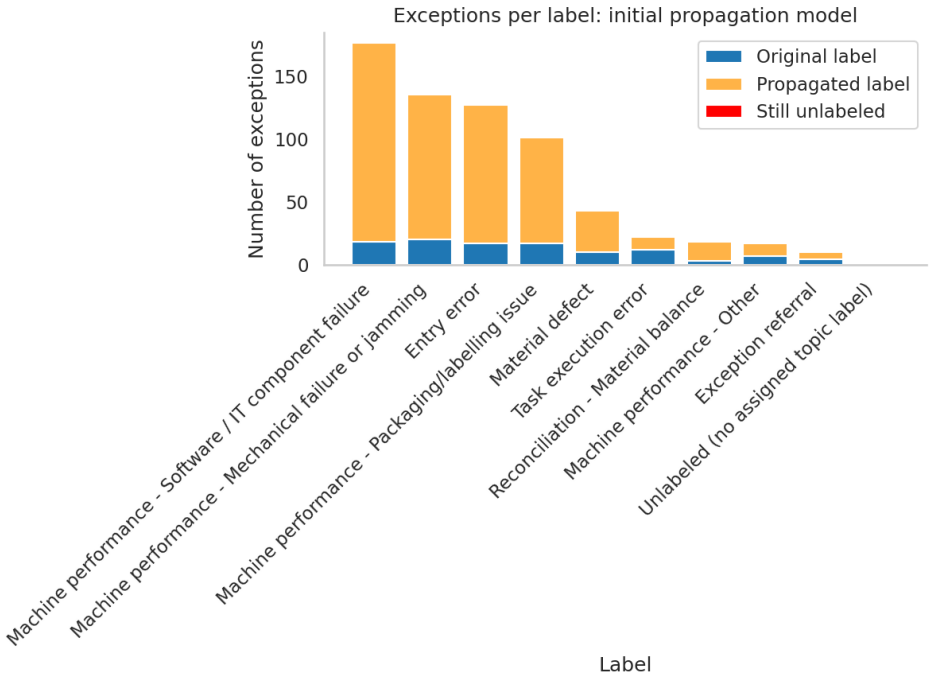

Figure 9: Original vs. propagated labels per class for the initial BERTopic propagation model

The only notable shift was the increase in the unlabeled portion of the dataset due to stricter clustering in the tuned model (3), although the confidence of the labels still increased (8).

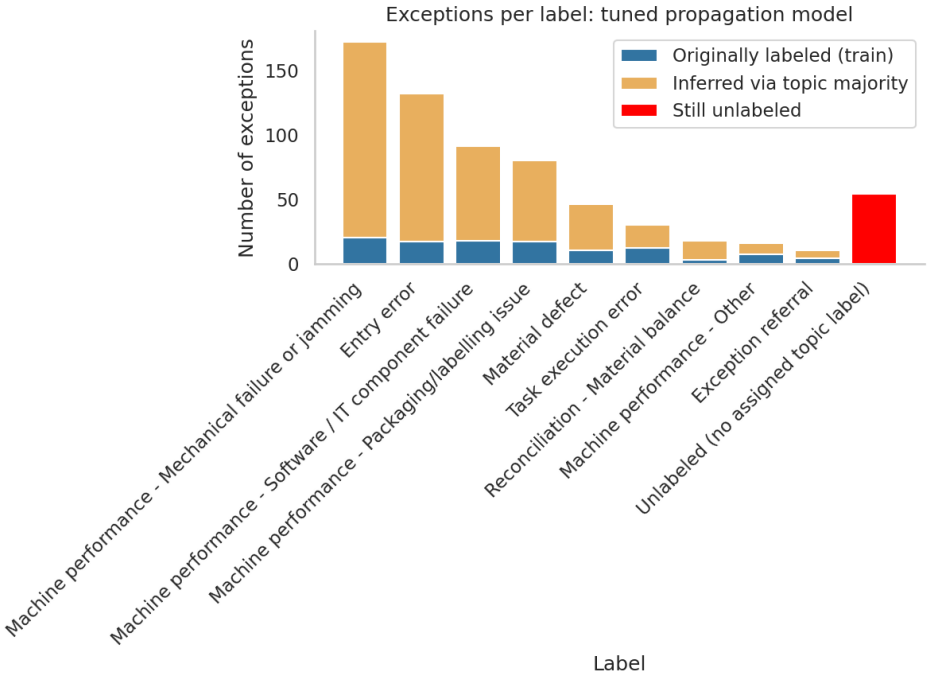

Figure 10: Original vs. propagated labels per class for the tuned BERTopic propagation model

The confusion matrices for the initial and tuned propagation models (Figures 11 and 12) help explain the gap between macro and micro F1 scores. Large classes, including Machine performance categories and Entry error, show the highest false positive rates.

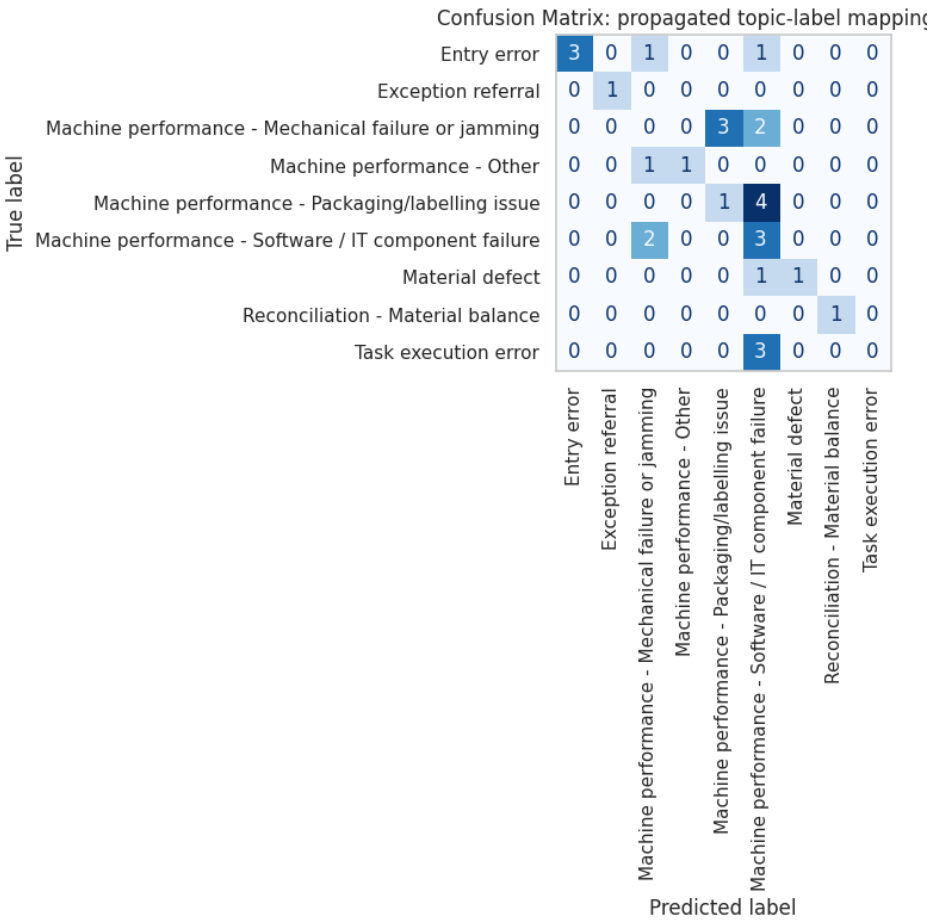

Figure 11: Initial BERTopic propagation model: Confusion matrix by class

In contrast, smaller classes such as Task execution error and Material Balance have fewer false positives. Since macro F1 weighs all classes equally, these smaller and cleaner classes contribute more to macro F1 than to micro F1, which is dominated by the performance of the majority classes.

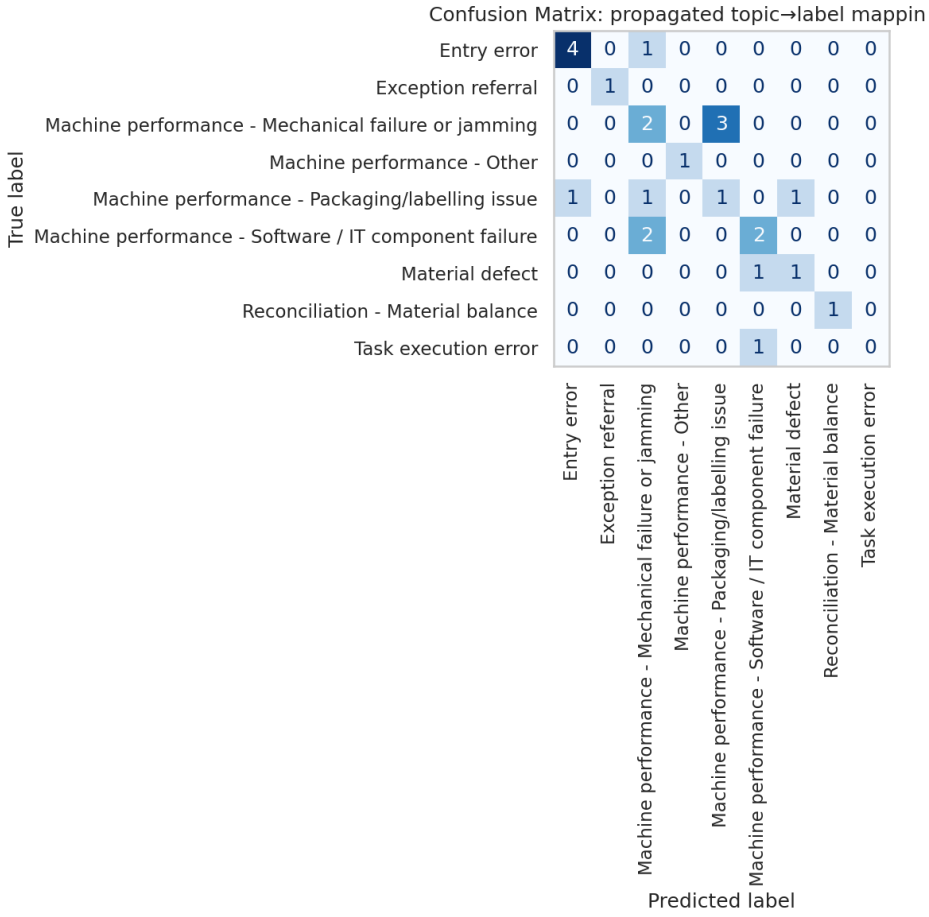

Figure 12: Tuned BERTopic propagation model: Confusion matrix

Despite these differences in macro vs. micro F1 scores and the slight decrease in propagated label coverage, the tuned BERTopic propagation model resulted in an increase in both accuracy and label confidence compared to the initial propagation model. Therefore, the propagated training data resulting from the tuned BERTopic propagation model is used to train the classifiers.

5.2 *Classifier training datasets*

Three variants of the training data were created from the tuned BERTopic label propagation to evaluate how the quantity of labels and confidence in the labels affect the learning of the classifier. The fully propagated dataset contained 595 labeled instances, of which 487 were propagated as shown in Table 4 below. Filtering out propagated labels with confidence below

0.6 produced the high confidence dataset with 538 instances, representing only a 10% reduction from the full post propagation set.

Table 4: Original vs. inferred labels for each training data variant

Dataset	Total Instances	Original Labels	Propagated Labels
Ground-truth labels only (gt)	108	108	0
Full propagation (postprop)	595	108	487
Original labels + high confidence propagated (highconf)	538	108	430

5.3 *Cross validation results*

Clear patterns emerged when comparing micro and macro CV F1 scores across training datasets and classifiers. Figure 13 shows the distribution of the mean CV F1 macro and micro scores for all combinations of hyperparameters tested. Both classifiers benefited from using propagated data. The high confidence (highconf) and fully propagated (postprop) datasets consistently produced higher CV macro and micro F1 scores than the ground truth only (gt) dataset. Additionally, models using the highconf and postprop datasets produced relatively similar F1 scores. This indicates that while reducing the amount of labeled data by selecting only high confidence points does not negatively affect accuracy, focusing solely on the high confidence points also does not substantially increase accuracy. LR produced more stable CV scores across variants, while RF showed greater variability, particularly in macro F1, reflecting sensitivity to class balance and decision boundary complexity. RF achieved overall higher micro and macro F1 scores in CV than LR for every dataset.

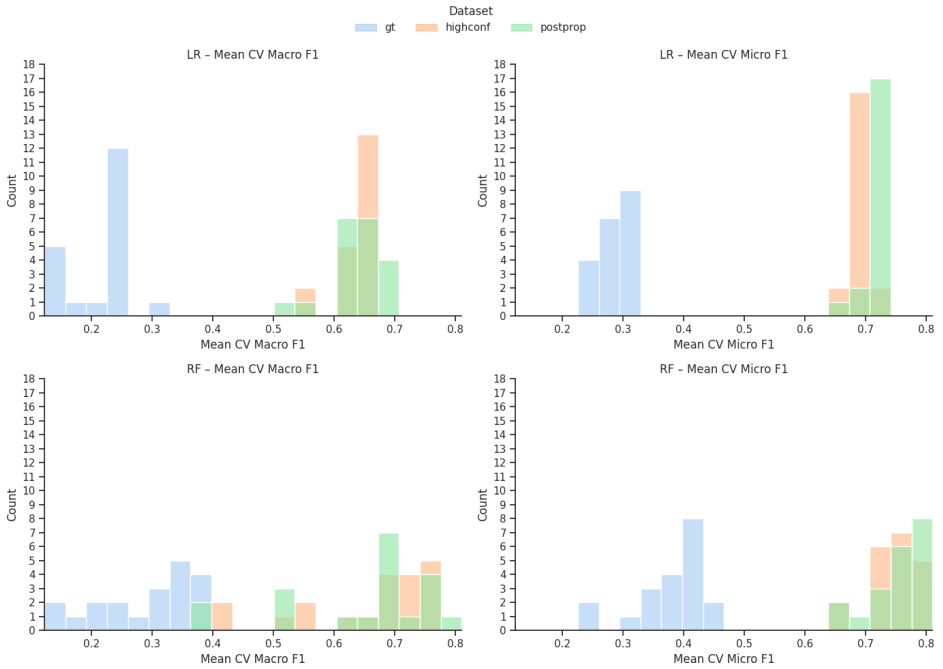

Figure 13: Distribution of cross-validation macro and micro F1 scores for Random Forest and Logistic Regression classifiers (LR = Logistic Regression, RF= Random Forest, gt= Original labeled dataset, highconf = High confidence dataset, postprop = Fully propagated dataset, CV= cross-validation)

5.4 *Test set results and generalization*

Four main findings emerge from evaluating the classifiers on the held-out test:

1. **CV F1 scores exceed test F1 scores by a large margin.** The fully propagated and high confidence datasets produced much stronger CV performance than the ground truth only dataset, as shown in Table 5. For example, CV macro F1 for RF reached 0.80 using the high confidence dataset and 0.79 using the fully propagated dataset, compared to 0.27 for the ground truth only dataset. The same pattern holds for LR. However, test set performance remained modest, with all models scoring below 0.60 for both macro and micro F1. The highest score for both classifiers was the macro F1 score, with RF achieving 0.56 and LR achieving 0.53, both trained on the high confidence dataset. This indicates that although the propagated labels support better learning, they still contain noise that limits generalization to unseen data. High CV accuracy coupled with low test set accuracy can indicate that the model is overfitting to the training data. However, it can

also indicate that classes with particularly heterogeneous or noisy content are making it difficult for the classifiers to properly draw a decision boundary, or that models are better at understanding which classes are more likely than at assigning the correct final label. This is explored further in the section Error Analysis below.

Table 5: Comparison of test and cross-validation macro and micro F1 scores for each training data variant (LogReg = Logistic Regression, gt = Original labeled dataset, highconf = High confidence dataset, postprop = Fully propagated dataset, CV = cross-validation)

Dataset	Classifier	Test Macro F1	Test Micro F1	Best CV Macro F1	Best CV Micro F1
gt	RandomForest	0.33	0.45	0.27	0.46
highconf	RandomForest	0.56	0.45	0.76	0.80
postprop	RandomForest	0.54	0.41	0.79	0.81
gt	LogReg	0.38	0.31	0.30	0.31
highconf	LogReg	0.53	0.45	0.66	0.71
postprop	LogReg	0.52	0.45	0.65	0.73

2. **Macro F1 scores exceed micro F1 scores on the test set.** In contrast to the CV results where micro F1 is generally higher, the test results show macro F1 as consistently higher than micro F1. For example, as shown in Table 5, the RF model trained on the high confidence dataset scored a macro F1 of 0.56 but a micro F1 of 0.45. This suggests that classifiers learn comparatively more reliable information about minority classes, while propagated majority class labels are less accurate or less consistent.

3. **Little information is lost by using the high confidence labels only for training.** Test set performance shows little difference in terms of macro and micro F1 scores between classifiers trained on the high confidence vs. the fully propagated training data, with the exception of the micro F1 for the LR classifier.

4. **RF outperforms LR in almost every case in terms of macro and micro F1.** As shown in Figure 14, which depicts the test set macro and micro F1 scores for both classifiers, the RF classifier consistently achieved higher macro and micro F1 scores for almost all training data variants. One exception is the macro F1 score for the original label-only dataset (gt), for which LR performed the best.

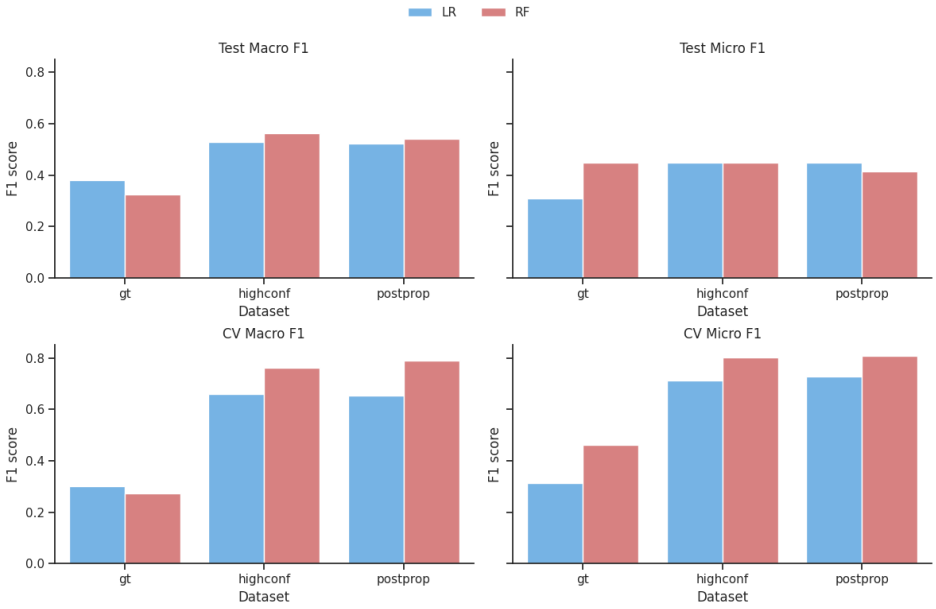

Figure 14: Visual comparison of test and cross-validation macro and micro F1 scores for each training data variant (LogReg = Logistic Regression, gt= Original labeled dataset, highconf = High confidence dataset, postprop = Fully propagated dataset, CV= cross-validation)

5.5 *Error analysis*

An analysis of error patterns for the best-performing model from each classifier in terms of test set macro F1 shows that each classifier underperforms in a similar pattern and on similar data points. The Receiver Operating Curves (ROC) curves for the best RF and LR models trained on the high-confidence dataset (Figures 15 and 16) confirm that both classifiers perform substantially better than random guessing and have learned to separate the classes of the dataset well. The best RF model achieves the highest overall Area Under Curve (AUC). Unlike the earlier F1 comparison, however, the AUC results show a notable difference between the two classifiers: for the RF model, the micro-average AUC is higher than the macro-average AUC, whereas for the LR model the opposite is true, with a higher macro-average AUC than micro-average AUC.

The combination of high AUC values yet relatively low F1 scores indicates that both classifiers have learned to distinguish embeddings from different classes in a ranking sense, but fail to translate those rankings into correct predictions. This contrast between AUC and F1 also indicates that the high CV F1 scores vs. the low test set F1 scores is likely due to particularly heterogeneous classes that make it difficult for the classifier to

distinguish and draw a clear decision boundary, rather than due solely to the model overfitting on the training data. For RF (Figure 15), the pattern of micro AUC > macro AUC suggests that it correctly ranks the most likely classes for a given data point and that it correctly ranks majority class examples more reliably than minority-class examples. However, the concurrent pattern of micro F1 < macro F1 shows that RF still struggles to convert those rankings into correct class assignments within the noisy majority classes created through label propagation.

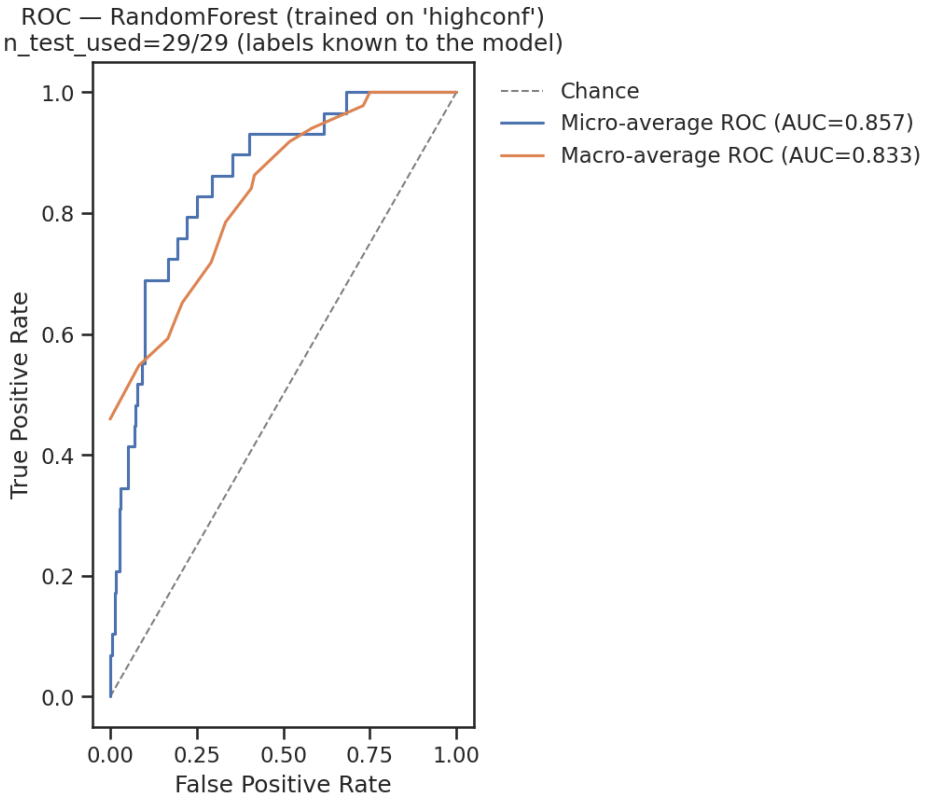

Figure 15: Receiver operating curve for the best Random Forest model, trained on the high confidence dataset (ROC = receiver operating curve, highconf = High confidence, AUC = area under curve)

In contrast, LR's macro AUC > micro AUC (Figure 16) indicates relatively stronger ranking performance on minority classes. Its micro-F1 is also lower than its macro F1 score, which shows that it also still struggles to make accurate decisions on the more frequent classes.

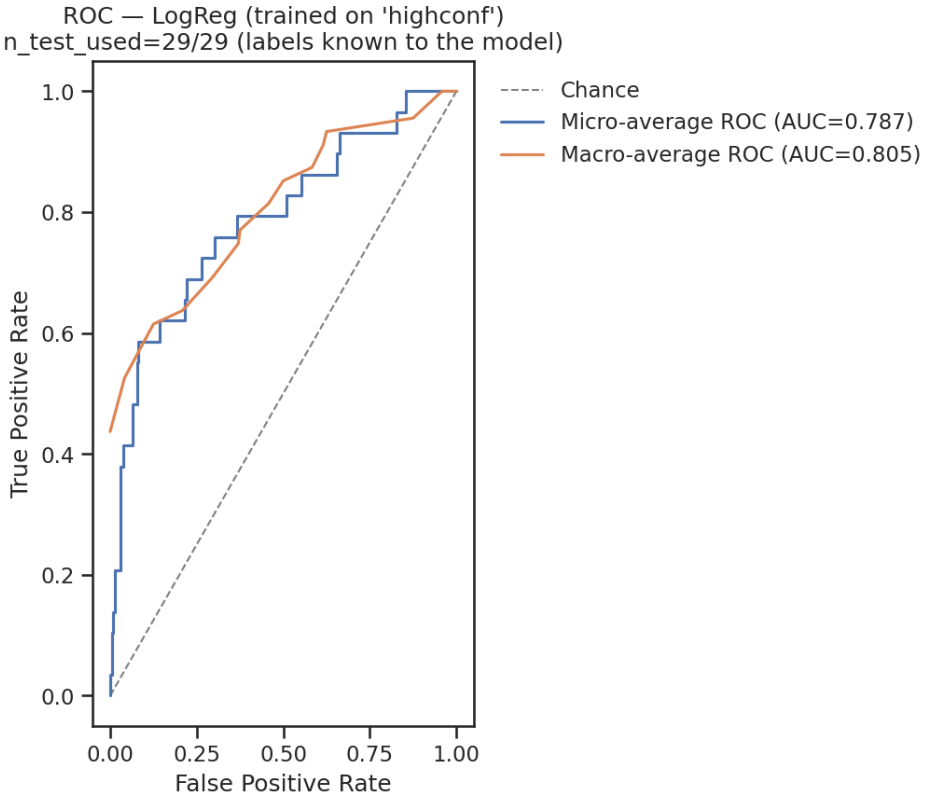

Figure 16: Receiver operating curve for the best Logistic Regression model, trained on the high confidence dataset (ROC = receiver operating curve, LogReg = Logistic Regression, highconf = High confidence, AUC = area under curve)

Table 6 below shows that the F1 scores for several of the largest classes, such as "Mechanical failure and jamming", "Packaging/labelling issue", and "Software / IT Component Failure" are considerably worse than all other classes, for both classifiers. These classes are all variations of machine performance related exceptions. This likely indicates that exceptions with these labels have increased variation in their embeddings and that the exceptions within these classes are too heterogeneous to be consistently and accurately classified. They may benefit from being broken down into further sub-categories or from extra text cleaning and standardization to make their user descriptions and embeddings more consistent.

Table 6: Classification report for Random Forest (RF) and Logistic Regression (LR) best models trained on the high confidence dataset

Class	RF Precision	RF Recall	RF F1	LR Precision	LR Recall	LR F1
Entry error	0.67	0.80	0.73	0.57	0.80	0.67
Exception referral	1.00	1.00	1.00	1.00	1.00	1.00
Mechanical failure or jamming	0.22	0.40	0.29	0.14	0.20	0.17
Machine performance – Other	1.00	0.50	0.67	1.00	0.50	0.67
Packaging/labelling issue	0.25	0.20	0.22	0.40	0.40	0.40
Software / IT component failure	0.33	0.20	0.25	0.50	0.40	0.44
Material defect	0.33	0.50	0.40	0.33	0.50	0.40
Reconciliation – Material balance	1.00	1.00	1.00	1.00	1.00	1.00
Task execution error	1.00	0.33	0.50	0.00	0.00	0.00
Accuracy			0.45			0.45
Macro Avg	0.65	0.55	0.56	0.55	0.53	0.53
Weighted Avg	0.52	0.45	0.45	0.44	0.45	0.43

Examining which classification errors each classifier has in common provides further insight. Table 7 indicates the number of errors produced by each classifier-dataset combination as well as errors that both classifiers have in common. RF and LR show substantial overlap in the errors they make when trained on both propagated training data variants (78% and 83% overlap). This suggests that certain data points or labels are consistently difficult for both classifiers.

Table 7: Number of errors unique and in common among different classifier and training dataset combinations (RF= Random Forest, LR= Logistic Regression, gt= Original labeled dataset, highconf = High confidence dataset, postprop = Fully propagated dataset)

Dataset	RF Errors	LR Errors	Either Errors	Both Errors	Overlap (Jaccard)
gt	16	20	24	12	0.50
highconf	16	16	18	14	0.78
postprop	17	16	18	15	0.83

6 DISCUSSION

The results show that automatic label propagation substantially enhances supervision for training classifiers on short, technical manufacturing text. Expanding the training set from 108 original labels to over 500 propagated labels enabled both RF and LR classifiers to learn far more stable decision boundaries, as reflected in the large increases in CV macro and micro F1 scores. The high-confidence dataset in particular produced the strongest and most consistent CV performance across classifiers. The tuned BERTopic label propagation model played a key role in this improvement. Tuning increased both micro and macro F1 scores on the held-out test

set and produced more coherent clusters, as shown by higher average membership strength and reduced variance. The tuned BERTopic propagation model achieved a macro F1 score of 0.60 using 20% labeled data, which is comparable to the result achieved by Babikov et al. (2023) who achieved an F1 score of 0.65 using a DBSCAN clustering algorithm and 25% labeled data. These gains suggest that BERTopic is capable of propagating labels with moderate accuracy using transformer-based embeddings and density-based clustering.

Even so, the remaining noise in the propagated labels, especially in the machine performance related categories, limited how effectively this supervision can generalize. Using only 20% labeled data was a practical necessity, given constraints on time and resources for collecting additional labeled data. Reducing the amount of unlabeled data to improve this ratio would have risked compromising the total dataset size, which is critical for effective classifier training. Future research could explore increasing the total training data size and the ratio of labeled to unlabeled data to match or exceed levels used in prior studies (Babikov et al., 2023), potentially further improving BERTopic's propagation accuracy. Despite these challenges, propagated supervision remains useful in this setting. Classifiers trained solely on the 108 original labels performed substantially worse than those trained on propagated datasets, demonstrating that the amount of original labels alone was not enough to train effective classifiers.

However, test set performance of the trained classifiers highlighted the limitations of the current approach. The high CV F1 scores achieved by both classifiers indicate that the models do indeed learn to distinguish classes, yet the low test set F1 scores indicate that the decision boundaries they learn are noisy and do not translate well to unseen data. Although both RF and LR achieved much higher CV F1 scores using propagated datasets, their test macro and micro F1 scores remained below 0.60. This is lower than classifiers trained on automatically propagated data from literature, such as the minimum F1 score of 0.65 achieved by Babikov et al.(2023) after using DBSCAN clustering for propagation and training an SBERT neural classifier. However, their more powerful neural classifier along with a higher proportion of labeled data may contribute to their higher F1 scores. The gap between CV and test scores suggests that while propagation helps classifiers learn the structure of propagated labels, inconsistencies remain in the labels of several large classes that constrains performance on the held-out test set. This aligns with the error patterns observed in the propagation models themselves. Larger classes continued to show high false-positive rates and low F1 scores in the final classification reports for each classifier (Table 6).

The ROC results further illustrate this dynamic (Figures 15 and 16). Both classifiers achieve strong AUC values, demonstrating that they can correctly rank classes even when they misclassify final labels. This pattern indicates that the embeddings contain meaningful semantic separation, but noise within classes distorts the final decision thresholds. Models are better at understanding which classes are more likely than at assigning the correct final label, particularly for larger classes, as indicated by the higher macro F1 than micro F1 for both models. Minority classes, on the other hand, showed fewer errors in the classification report (Table 6). However, the total number of samples for minority classes remained small in absolute numbers, both from original and propagated labels. Future work should aim to attain labeled data with a more even distribution of class sizes to better assess which classes contain the most variation when controlling for size. Finally, the RF classifier consistently outperformed LR across nearly all datasets, particularly on the propagated ones. RF's higher macro and micro F1 scores, along with its superior AUC, suggest that it is better suited to handling variability and nonlinear decision boundaries created through weak supervision. Overall, remaining noise and class heterogeneity, especially within larger classes, limited generalization and highlighted areas where refined label definitions or selective human supervision would further strengthen performance. Future work should explore whether creating more fine-grained categories that can break up noisy majority classes can provide better generalization capability. Although the classifiers trained here did not yet reach a level of accuracy to reliably classify MES exceptions for root-cause analysis and improving industrial reliability, this methodology shows strong potential for this purpose with additional tailoring.

7 CONCLUSION

This thesis investigated whether semi-supervised topic modeling via BERTopic automatic label propagation could improve classification of short, technical manufacturing text using RF and LR classifiers. Motivated by limited labeled data and the need for scalable supervision, the study addressed a gap in machine-learning applications for MES exceptions. Results demonstrated that label propagation significantly expanded the training signal. The tuned BERTopic model achieved accurate propagation, as evaluated on the held-out test set, confirming that transformer-based embeddings with density-based clustering can extend supervision in low-label scenarios. Expanding the dataset from 108 human labels to over 500 propagated labels improved cross-validated macro and micro F1 scores for both classifiers, with RF consistently outperforming LR.

However, evaluation on the test set revealed limitations. The best models achieved macro F1 scores of only 0.52–0.56, with even lower micro F1, suggesting persistent ambiguity in certain exception types. Noise in propagated labels, particularly in majority classes, likely constrained generalization. While propagation enhances training, it does not fully resolve inconsistencies in the label taxonomy or embedding variability. Future improvements could include increasing labeled data quantity and balance, or targeted human review for noisy categories like machine performance exceptions. Despite these challenges, the findings confirm that BERTopic-based label propagation offers a scalable solution for text classification when extensive manual labeling is impractical.

REFERENCES

Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., & Hassan, A. (2023). Topic modeling algorithms and applications: A survey. *Information Systems*, *112*, 102131. https://doi.org/10.1016/j.is.2022.102131

AI, O. (2025). Text-embedding-large-3. Retrieved November 20, 2025, from https://platform.openai.com

Angelov, D. (2020, August). Top2Vec: Distributed Representations of Topics [arXiv:2008.09470 [cs]]. https://doi.org/10.48550/arXiv.2008.09470

Asas, V., Juan, S. S., Kerbun, V. C., Chua, S., Labadin, J., & Lau, E. (2025). A Comparative Analysis of Topic Modelling Techniques for Malaysian Business News Data: LDA, NMF, Top2Vec, and BERTopic. *2025 14th International Conference on Information Technology in Asia (CITA)*, 42–47. https://doi.org/10.1109/CITA66455.2025.11198734

Babikov, I., Kovalchuk, S., & Soldatov, I. (2023). Semi-supervised method for improving general-purpose and domain-specific textual corpora labels. *Procedia Computer Science*, *229*, 168–176. https://doi.org/10.1016/j.procs.2023.12.018

Berkson, J. (1944). Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association*, *39*(227), 357–365. https://doi.org/10.1080/01621459.1944.10500699

Bianchi, F., Terragni, S., & Hovy, D. (2021). Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 759–766. https://doi.org/10.18653/v1/2021.acl-short.96

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*, 106. https://doi.org/10.1186/1471-2105-14-106

Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of Science [arXiv:0708.3601 [stat]]. *The Annals of Applied Statistics*, *1*(1). https://doi.org/10.1214/07-AOAS114

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022. https://dl.acm.org/doi/10.5555/944919.944937

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Bu, W., Shu, H., Kang, F., Hu, Q., & Zhao, Y. (2023). Software Subclassification Based on BERTopic-BERT-BiLSTM Model [Publisher: Multidisciplinary Digital Publishing Institute]. *Electronics*, *12*(18), 3798. https://doi.org/10.3390/electronics12183798

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J., & Blei, D. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems*, *22*. Retrieved October 23, 2025, from https://proceedings.neurips.cc/paper_files/paper/2009/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html

Churchill, R., & Singh, L. (2022). The Evolution of Topic Modeling. *ACM Comput. Surv.*, *54*(10s), 215:1–215:35. https://doi.org/10.1145/3507900

Dieng, A. B., Ruiz, F. J. R., & Blei, D. M. (2019, October). The Dynamic Embedded Topic Model [arXiv:1907.05545 [cs]]. https://doi.org/10.48550/arXiv.1907.05545

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology*, *38*(1), 188–230. https://doi.org/10.1002/aris.1440380105

Egger, R., & Yu, J. (2022). A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts [Publisher: Frontiers]. *Frontiers in Sociology*, *7*. https://doi.org/10.3389/fsoc.2022.886498

Gan, L., Yang, T., Huang, Y., Yang, B., Luo, Y. Y., Richard, L. W. C., & Guo, D. (2024). Experimental Comparison of Three Topic Modeling Methods with LDA, Top2Vec and BERTopic. In H. Lu & J. Cai (Eds.), *Artificial Intelligence and Robotics* (pp. 376–391). Springer Nature. https://doi.org/10.1007/978-981-99-9109-9_37

Garbe, W. (2025). SymSpell [original-date: 2014-03-25T11:01:35Z]. Retrieved November 20, 2025, from https://github.com/wolfgarbe/SymSpell

Gottumukkala, D. P., P. V.g.d, P. R., & Rao, S. K. (2025). Topic modeling-based prediction of software defects and root cause using BERTopic, and multioutput classifier [Publisher: Nature Publishing Group]. *Scientific Reports*, *15*(1), 25428. https://doi.org/10.1038/s41598-025-11458-0

Grootendorst, M. (2022, March). BERTopic: Neural topic modeling with a class-based TF-IDF procedure [arXiv:2203.05794 [cs]]. https://doi.org/10.48550/arXiv.2203.05794

Hankar, M., Kasri, M., & Beni-Hssane, A. (2025). A comprehensive overview of topic modeling: Techniques, applications and challenges. *Neurocomputing*, *628*, 129638. https://doi.org/10.1016/j.neucom.2025.129638

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*.

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). Spacy: Industrial-strength natural language processing in python. https://doi.org/10.5281/zenodo.1212303

Huseynova, K., & Isbarov, J. (2024, August). Enhanced document retrieval with topic embeddings [arXiv:2408.10435 [cs]]. https://doi.org/10.48550/arXiv.2408.10435

Kazanci, N. (2025). Extended topic classification utilizing LDA and BERTopic: A call center case study on robot agents and human agents. *Applied Intelligence*, *55*(5), 360. https://doi.org/10.1007/s10489-024-06106-5

Kim, J., & Lee, S. (2024). Technology Opportunity Analysis for Creating Innovative Solutions: Applying Semi-supervised Topic Modelling on Patent Data [ISSN: 2159-5100]. *2024 Portland International Conference on Management of Engineering and Technology (PICMET)*, 1–9. https://doi.org/10.23919/PICMET64035.2024.10653159

Laureate, C. D. P., Buntine, W., & Linger, H. (2023). A systematic review of the use of topic models for short text social media analysis. *Artificial Intelligence Review*, *56*(12), 14223–14255. https://doi.org/10.1007/s10462-023-10471-x

Li, C., Duan, Y., Wang, H., Zhang, Z., Sun, A., & ma, Z. (2017). Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems*, *36*, 1–30. https://doi.org/10.1145/3091108

Li, X., Zhang, A., Li, C., Guo, L., Wang, W., & Ouyang, J. (2019). Relational Biterm Topic Model: Short-Text Topic Modeling using Word Embeddings. *The Computer Journal*, *62*(3), 359–372. https://doi.org/10.1093/comjnl/bxy037

Li, Y., Yang, Y., Song, P., Duan, L., & Ren, R. (2025). An improved SMOTE algorithm for enhanced imbalanced data classification by expanding sample generation space [Publisher: Nature Publishing Group]. *Scientific Reports*, *15*(1), 23521. https://doi.org/10.1038/s41598-025-09506-w

Ma, L., Chen, R., Ge, W., Rogers, P., Lyn-Cook, B., Hong, H., Tong, W., Wu, N., & Zou, W. (2025). AI-powered topic modeling: Comparing LDA and BERTopic in analyzing opioid-related cardiovascular risks in women. *Experimental Biology and Medicine*, *250*, 10389. https://doi.org/10.3389/ebm.2025.10389

McInnes, L., Healy, J., & Astels, S. (2017). Hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205. https://doi.org/10.21105/joss.00205

McInnes, L., Healy, J., & Melville, J. (2020, September). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction [arXiv:1802.03426 [stat]]. https://doi.org/10.48550/arXiv.1802.03426

Mekala, D., & Shang, J. (2020). Contextualized Weak Supervision for Text Classification. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 323–333. https://doi.org/10.18653/v1/2020.acl-main.30

Miao, Y., Yu, L., & Blunsom, P. (2016, June). Neural Variational Inference for Text Processing [arXiv:1511.06038 [cs]]. https://doi.org/10.48550/arXiv.1511.06038

Mishra, M. (2024). A holistic review of customer experience research: Topic modelling using BERTopic. *Marketing Intelligence & Planning*, *43*(4), 802–820. https://doi.org/10.1108/MIP-09-2023-0457

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic Evaluation of Topic Coherence. Retrieved October 23, 2025, from https://www.semanticscholar.org/paper/Automatic-Evaluation-of-Topic-Coherence-Newman-Lau/8e31f3c7e70e9a5f8afafd86cebc004d5eca8c2b

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*(2), 111–126. https://doi.org/10.1002/env.3170050203

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://arxiv.org/abs/1201.0490

Qiang, J., Qian, Z., Li, Y., Yuan, Y., & Wu, X. (2022). Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1427–1445. https://doi.org/10.1109/TKDE.2020.2992485

Ravenda, F., Bahrainian, S. A., Raballo, A., Mira, A., & Crestani, F. (2025). A self-supervised seed-driven approach to topic modelling and

clustering. *Journal of Intelligent Information Systems*, *63*(1), 333–353. https://doi.org/10.1007/s10844-024-00891-8

Rezaei, M. R., Hafezi, M., Satpathy, A., Hodge, L., & Pourjafari, E. (2024, October). AT-RAG: An Adaptive RAG Model Enhancing Query Efficiency with Topic Filtering and Iterative Reasoning [arXiv:2410.12886 [cs]]. https://doi.org/10.48550/arXiv.2410.12886

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 399–408. https://doi.org/10.1145/2684822.2685324

Rüdiger, M., Antons, D., Joshi, A. M., & Salge, T.-O. (2022). Topic modeling revisited: New evidence on algorithm performance and quality metrics [Publisher: Public Library of Science]. *PLOS ONE*, *17*(4), e0266325. https://doi.org/10.1371/journal.pone.0266325

Sabbagh, R., & Ameri, F. (2019). A Framework Based on K-Means Clustering and Topic Modeling for Analyzing Unstructured Manufacturing Capability Data. *Journal of Computing and Information Science in Engineering*, *20*(011005). https://doi.org/10.1115/1.4044506

Sala, R., Francalanza, E., & Arena, S. (2025). A review on three decades of manufacturing maintenance research: Past, present and future directions [Publisher: Taylor & Francis]. *Production & Manufacturing Research*, *13*(1), 2469037. https://doi.org/10.1080/21693277.2025.2469037

Söderwall, A., & Telešova, G. (2025). A Comparative Study with LDA and BERTopic: AI Policies Across Different Democracy Indexes [Accepted: 2025-10-06T13:34:43Z]. Retrieved October 21, 2025, from https://gupea.ub.gu.se/handle/2077/89821

Srivastava, A., & Sutton, C. (2017). AUTOENCODING VARIATIONAL INFERENCE FOR TOPIC MODELS. *ICLR 2017*. https://arxiv.org/abs/1703.01488

Vayansky, I., & Kumar, S. A. P. (2020). A review of topic modeling methods. *Information Systems*, *94*, 101582. https://doi.org/10.1016/j.is.2020.101582

Wang, J., & Hsu, C.-C. (2020). A topic-based patent analytics approach for exploring technological trends in smart manufacturing. *Journal of Manufacturing Technology Management*, *32*(1), 110–135. https://doi.org/10.1108/JMTM-03-2020-0106

Wang, Y., Benavides, R., Diatchenko, L., Grant, A. V., & Li, Y. (2022). A graph-embedded topic model enables characterization of diverse pain phenotypes among UK biobank individuals. *iScience*, *25*(6), 104390. https://doi.org/10.1016/j.isci.2022.104390

Wankmüller, S. (2023). A comparison of approaches for imbalanced classification problems in the context of retrieving relevant documents for an analysis. *Journal of Computational Social Science*, *6*(1), 91–163. https://doi.org/10.1007/s42001-022-00191-7

Xiong, H., Cheng, Y., Zhao, W., & Liu, J. (2019). Analyzing scientific research topics in manufacturing field using a topic model. *Computers & Industrial Engineering*, *135*, 333–347. https://doi.org/10.1016/j.cie.2019.06.010

Xu, S., Wang, Y., Cheng, X., & Yang, Q. (2025). Thematic Identification Analysis of Equipment Quality Problems Based on the BERTopic Model [ISSN: 2352-5428], 484–491. https://doi.org/10.2991/978-94-6463-676-5_47

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. *Proceedings of the 22nd international conference on World Wide Web*, 1445–1456. https://doi.org/10.1145/2488388.2488514

Zhang, Z.-W., Jing, X.-Y., & Wang, T.-J. (2017). Label propagation based semi-supervised learning for software defect prediction. *Automated Software Engineering*, *24*(1), 47–69. https://doi.org/10.1007/s10515-016-0194-x

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Scholkopf, B. (2004). Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems*. Retrieved November 5, 2025, from https://dl.acm.org/doi/10.5555/2981345.2981386

Zhou, Y., Lin, H., Liu, Y., & Ding, W. (2019). A novel method to identify emerging technologies using a semi-supervised topic clustering model: A case of 3D printing industry. *Scientometrics*, *120*(1), 167–185. https://doi.org/10.1007/s11192-019-03126-8

APPENDIX A

This appendix contains additional information, figures and tables related to the datasets and text of the thesis. The following Excel files containing data samples used in the thesis are included with the thesis submission. Please refer to Figure 4 for the sampling process. Certain columns mentioned in Table 1 are not included in the Excel files, such as the raw userdescription attribute, given that they contain confidential and proprietary information belonging to Amgen Breda.

Initialsample_DSS_revised.xlsx: This file contains the main sample of 9,989 rows taken from the full dataset upon which text cleaning, standardization and transformations were completed before drawing the unlabeled and labeled samples.

Unlabeledsample_DSS_revised.xlsx: This file contains the unlabeled sample of 541 rows. The labeled sample was appended to this sample to perform BERTopic label propagation

Labeledsample_DSS_revised.xlsx: This file contains the sample that was labeled by domain experts from Amgen Breda. These are the labeled data points that acted as weak supervision for BERTopic label propagation.

7.1 *Dataset characteristics*

Table 8 below gives an extended example of the user descriptions used in the analysis and how they look after text cleaning and standardization are applied.

Table 8: Example exception data with selected attributes

Exception Number	**User Description**
6748266006	During the Set up phase on segment [NUMBER], [PERSON] was unable to scan the IDP batch number on IDP shippers according to [FORM SOP] [NUMBER] [IDENTIFIER].[NUMBER] step [NUMBER]. [NOTIFIED] [PERSON] performed a [NUMBER]% check of the IDP batch number on every IDP shipper using the pick report. Result is pass.
6811191596	Today at prep [IDENTIFIER] for PO [NUMBER] we received [IDENTIFIER] of flutes. We expected a shipper of [IDENTIFIER] of batch [NUMBER] with material number [NUMBER], actually received is [IDENTIFIER]. [PERSON] is [NOTIFIED] and [PERSON] is [NOTIFIED]. We can run the order with [IDENTIFIER] of flutes. Line will scrap [IDENTIFIER] of batch [NUMBER]. WHS is [NOTIFIED].
6838590153	"[PERSON] What: Can bus error Where: partner When: During production, Partner counter: [NUMBER] Actions taken: took pictures and was able to reset the can bus error with his account. Followed: Followed [NUMBER] on flow, no issues occurred. Continue production."

Figures 17, 18, 19, and 20 below illustrate different facets of the dataset used in the study. Figures 17 and 18 show that dropping exceptions with

user descriptions with a character length less than 11 did not substantially change the overall distribution of the character length for the data overall.

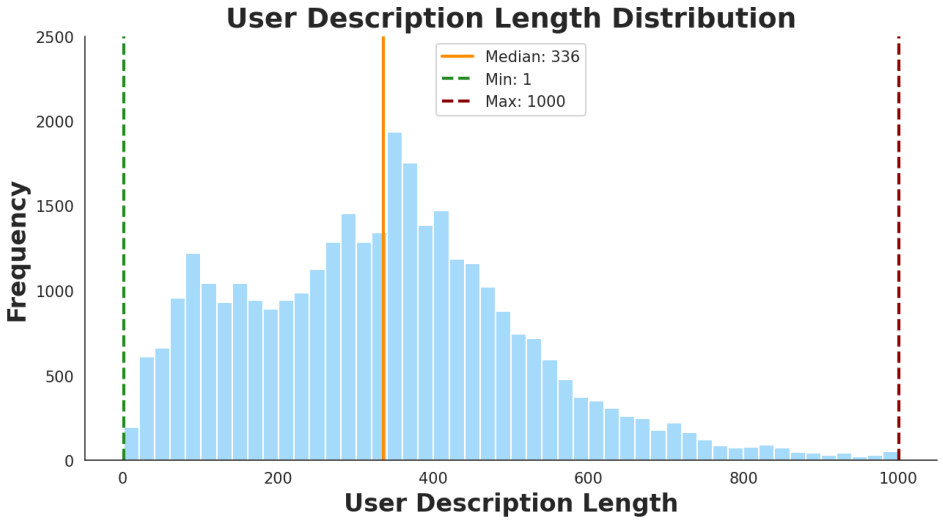

Figure 17: Distribution of user description lengths before dropping at thresholds len < 11

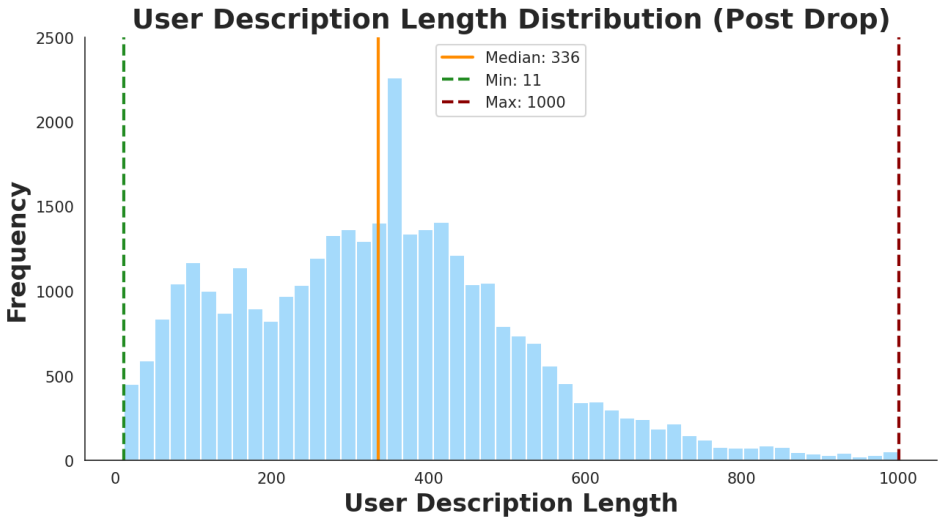

Figure 18: Distribution of user description lengths after dropping at thresholds len < 11

Figure 19 shows the proportion of different event types in the original dataset. This attribute was chosen to stratify the labeled and unlabeled samples given the highly unbalanced nature of the attribute and its link to the types of content included in the user description.

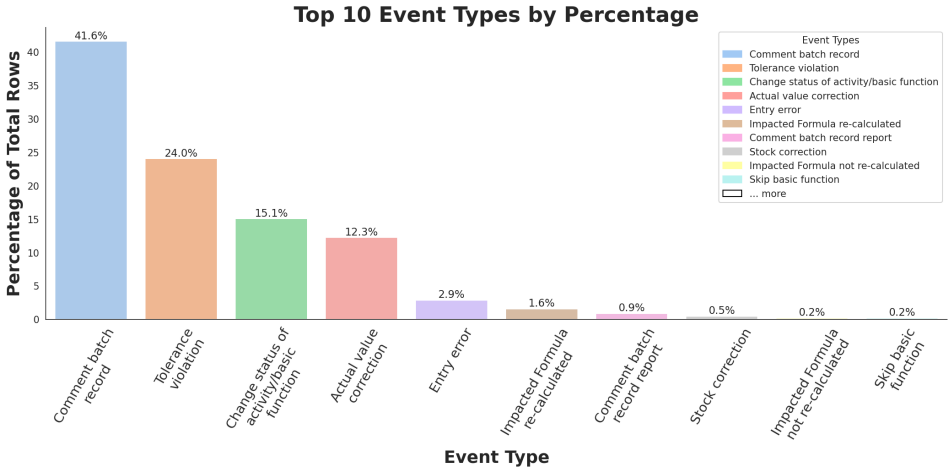

Figure 19: Proportion of exceptions within each event types

Figure 20 below shows the distribution of dates for which exceptions were registered in the original dataset. The date distribution is relatively uniform, with no strong skew towards either earlier or more recent time periods. Therefore, no stratification was performed during the sampling based on the date attribute.

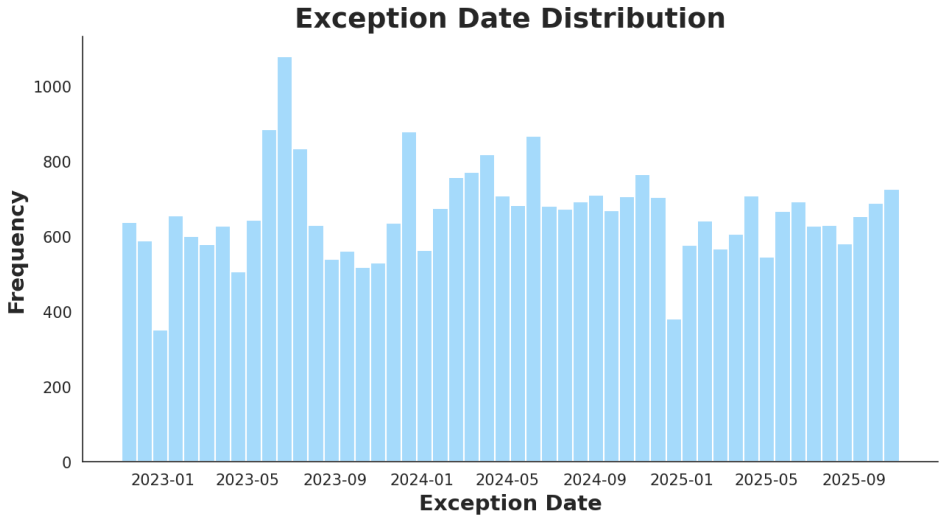

Figure 20: Distribution of exceptions by date

7.2 *Data preprocessing*

Additional detailed information on the different data preprocessing steps can be found in the sections below. Pseudocode representing the text cleaning and standardization pipeline can be found in Appendix B.

7.2.1 *User description text cleaning*

The user description attribute was first cleaned to remove person names and role tags (Operator, QA, Sr. Ass.). The names of the employees were replaced with a placeholder [PERSON] to preserve the semantic relationship of the sentence when embedded. Similarly, all references to specific forms or standard operating procedures (SOPs), as well as numbers and alphanumeric strings, were removed and replaced with similar placeholders to preserve semantic relationships. Phrases such as "notified" and "informed" plus a person's name were replaced with the placeholder [INFORMED]. This type of phrase was very repetitive in the user descriptions, as operators are often required to notify quality assurance personnel about exceptions and their resolutions. Therefore, this placeholder was used to standardize the presence of this type of text content throughout the exceptions. Special punctuation marks (i.e.) , (, +, /) were removed, while punctuation marks holding semantic meaning (i.e. &) were replaced with their corresponding word. Similar and frequently occurring phrases holding special interest for manufacturing were standardized for consistency in the text cleaning (i.e "wrong entry", "incorrect entry", "error entry" and "entry error" were all standardized to "entry error" for consistency).

7.2.2 *Labeling process*

A sample of maximum 150 rows, stratified by event type, was taken from the initial sample. Due to rounding in the event type stratification, this resulted in a total of 144 exceptions. These 144 exceptions were divided into 9 batches of 15 exceptions each and 1 batch of 9 exceptions. Each batch was given to a domain expert in the manufacturing or quality assurance departments. Instructions were given for the domain experts to label each exception with the category that they felt best fit the content of the exception's user description. The respondents received an initial list of 9 potential categories to choose from that was developed in consultation with the management staff. If they did not feel that any of the 9 predefined categories applied, they were instructed to write in their own category. After each batch was labeled and returned, the results were condensed and all categories that were not on the predefined list were reviewed and consolidated based on their commonalities. Some new categories were filled in by the respondents, while some of the categories provided in the predefined list were not used. In the end, 11 categories resulted from the labeling process. Since 3 exceptions were left uncategorized due to difficulty in classifying, a total of 141 labeled exceptions resulted from

the labeling process. Table 9 below shows the number of batches and the number of labeled exceptions per batch.

Table 9: Results of labeling process

Batch	N. exceptions to label	N. left blank
1	15	1
2	15	2
3	15	0
4	15	0
5	15	0
6	15	0
7	15	0
8	15	0
9	15	0
10	9	0

7.2.3 *Labeled train-test split*

The 141 rows of the labeling sample were split into a training set and a test set. Once all domain experts' labels were collected, the labeled sample was split into train and test sets using an 80/20 split and stratified by the label class, with the guardrail that at least one instance of each class must be present in both the train and the test set. This was done to ensure that the trained model was capable of predicting every class. However, the result showed that two classes only had 1 sample each in the train and test sets, the classes "Reconciliation - Label reconciliation" and the class "General - Other". In anticipation of the problems this would cause for the later stratified K-fold cross validation procedure for tuning the propagation and classifiers, as well as the negative effect such rare classes would play in the resulting F1 score, these classes were dropped. This allowed both the propagation and classifier training to focus on learning and accurately predicting those classes for which there was sufficient information, at the expense of losing the ability to predict rare classes. An 80/20 split was used to create train and test sets due to the relative scarcity of labeled points vs. the availability of unlabeled data, as leaving 80% of the labeled points in the train set allowed for more supervision signal for label propagation and for the stratified K-fold cross validation used in hyperparameter tuning. This resulted in 108 labeled exceptions used as training data and 29 labeled exceptions used as a held-out test set, each with 9 classes.

7.3 *Label propagation*

The tables in this section give more detailed information on the hyperparameters used in each BERTopic propagation model as well as the resulting class distributions after label propagation with each model. Tables 10, 11, and 12 show the default parameters for the initial BERTopic propagation models, class distribution resulting after propagation and the classification report overall and by class, respectively, for the initial BERTopic propagation model.

Table 10: Default UMAP and HDBSCAN parameters in `BERTopic`

Component	Parameter	Default value / Description
UMAP	n_neighbors	15
	n_components	5
	min_dist	0.0
	metric	`cosine`
	random_state	42
HDBSCAN	min_cluster_size	10
	min_samples	None (defaults to min_cluster_size)
	metric	`euclidean`
	cluster_selection_method	`eom` (Excess of Mass)
	prediction_data	True (enables topic probabilities)

Table 11: Initial BERTopic propagation model class counts

Label name	Original labeled	Inferred from topics	Total
Machine performance - Software / IT component failure	18	158	176
Machine performance - Mechanical failure or jamming	20	115	135
Entry error	17	110	127
Machine performance - Packaging/labelling issue	17	84	101
Material defect	10	33	43
Task execution error	12	10	22
Reconciliation - Material balance	3	15	18
Machine performance - Other	7	10	17
Exception referral	4	6	10
Unlabeled (no assigned topic label)	0	0	0
Total	**108**	**541**	**649**

Table 12: Initial propagation model: Classification report

Class name	Precision	Recall	F1-score	Support
Entry error	1.00	0.60	0.75	5
Exception referral	1.00	1.00	1.00	1
Machine performance – Mechanical failure or jamming	0.00	0.00	0.00	5
Machine performance – Other	1.00	0.50	0.67	2
Packaging/labelling issue	0.25	0.20	0.22	5
Software / IT component failure	0.21	0.60	0.32	5
Material defect	1.00	0.50	0.67	2
Reconciliation – Material balance	1.00	1.00	1.00	1
Task execution error	0.00	0.00	0.00	3

Tables 13, 14, and 15 below show the hyperparameters used in tuning the BERTopic propagation model, the resulting class distribution after propagation with the tuned model and resulting classification report for the tuned BERTopic propagation model, respectively.

Table 13: Parameter distributions used for BERTopic hyperparameter tuning

Component	Parameter	Values Tested
UMAP	umap_model__n_neighbors	[5, 10, 15, 20]
	umap_model__n_components	[2, 5, 10, 15]
	umap_model__min_dist	[0.0, 0.01, 0.05, 0.1]
	umap_model__metric	[“cosine”]
	nr_topics	[None]
HDBSCAN	hdbscan_model__min_cluster_size	[5, 10, 15, 20]
	hdbscan_model__min_samples	[None, 5, 10, 15]
	hdbscan_model__cluster_selection_method	[“eom”]
	nr_topics	[None]
Topic Reduction	nr_topics_grid	[10, 15, 20, 30]

Table 14: Tuned BERTopic propagation model class counts

Label name	Original labeled	Inferred from topics	Total
Machine performance - Mechanical failure or jamming	20	152	172
Entry error	17	115	132
Machine performance - Software / IT component failure	18	73	91
Machine performance - Packaging/labelling issue	17	63	80
Material defect	10	36	46
Task execution error	12	18	30
Reconciliation - Material balance	3	15	18
Machine performance - Other	7	9	16
Exception referral	4	6	10
Unlabeled	0	54	54
TOTAL	**108**	**541**	**649**

Table 15: Tuned propagation model: Classification report

Class name	Precision	Recall	F1-score	Support
Entry error	0.80	0.80	0.80	5
Exception referral	1.00	1.00	1.00	1
Machine performance – Mechanical failure or jamming	0.33	0.40	0.36	5
Machine performance – Other	1.00	1.00	1.00	1
Machine performance – Packaging/labeling issue	0.25	0.25	0.25	4
Machine performance – Software/IT component failure	0.50	0.50	0.50	4
Material defect	0.50	0.50	0.50	2
Reconciliation – Material balance	1.00	1.00	1.00	1
Task execution error	0.00	0.00	0.00	1

7.4 *Classifiers*

This section includes more detailed information about the hyperparameter tuning process of classifiers, including the hyperparameter grid used for each classifiers, the architecture used to run the tuning process, the resulting training data variants produced and the configuration and results for the best models for each classifier.

A finite subset of hyperparameter combinations for each classifier was generated once using scikit-learn's `ParameterSampler` with a fixed random seed, guaranteeing that identical parameter sets were evaluated on every dataset. `GridSearchCV` from scikit-learn was used to test each combination of model and dataset in the fixed subset of hyperparameter combinations using macro F1 as the optimization metric. For the LR models, a `StandardScaler` step was used to standardize all embedding dimensions before fitting the model to ensure smooth convergence

of the solver along the loss function. Each classifier was implemented as an `imblearn pipeline`. The RF models were trained using scikit-learn's `RandomForestClassifier` while LR models were trained using scikit-learn's `LogisticRegression`.

The following tables provide extra information about the training dataset variants used in training the classifiers and the hyperparameters and final model configurations produced from tuning. Table 16 shows the hyperparameters used in `GridSearchCV` with a stratified three-fold cross validation process for each classifier.

Table 16: Hyperparameter grids for Random Forest (RF) and Logistic Regression (LR) classifiers used in GridSearchCV

Classifier	Hyperparameters Tested	
RF	*n_estimators*	[300, 400, 500, 600, 700, 800]
	max_depth	[4, 6, 8, 12, 16]
	min_samples_split	[2, 4, 8, 16]
	min_samples_leaf	[2, 4, 8, 10]
	max_features	["sqrt", 0.5]
	bootstrap	[True]
	class_weight	[None, "balanced", "balanced_subsample"]
LR	*solver*	["lbfgs", "saga"]
	penalty	["l2"]
	C	[0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0]
	class_weight	[None, "balanced"]
	multi_class	["multinomial", "ovr"]

Tables 17 and 18 below show the best model per training data variant for the RF and LR classifiers, respectively, and their hyperparameters.

Table 17: Random Forest (RF) best model configuration per dataset variant (gt = Original labels only, highconf = Original labels + high confidence labels only, postprop = All original and propagated labels)

Dataset	Clf	Test MaF1	Test MiF1	CV MaF1	CV MiF1	Boot	Class Wt	Depth	Feat	Est
gt	RF	0.33	0.45	0.27	0.46	true	null	4	0.5	700
highconf	RF	0.56	0.45	0.76	0.80	true	balanced_subsample	6	sqrt	800
postprop	RF	0.54	0.41	0.79	0.81	true	balanced	6	sqrt	500

Table 18: Logistic Regression (LR) best model configuration per dataset variant (gt = Original labels only; highconf = Original labels + high confidence labels only; postprop = All original and propagated labels)

Dataset	Clf	Test MaF1	CV MaF1	Class Wt	C	MultiCls	Penalty	Solver
gt	LR	0.38	0.30	balanced	2.00	multinomial	l2	saga
highconf	LR	0.55	0.69	balanced	0.01	multinomial	l2	saga
postprop	LR	0.54	0.68	balanced	0.01	multinomial	l2	saga

APPENDIX B

This appendix contains all information related to the code and any custom algorithms used in the execution of the thesis.

7.5 *Code files*

Different parts of the thesis were executed in separate code files. Each of the code files is listed below with a brief explanation of the parts of the thesis they contain.

DataPreparation.ipynb: This file contains all code related to initial exploration of the dataset, summary statistics, building the text cleaning and standardization pipeline and applying the text cleaning to both the initial and unlabeled samples.

Main.ipynb: This code file contains all code related to BERTopic label propagation and training the downstream classifiers, including importing the unlabeled and labeled samples, training and hyperparameter tuning pipelines and error analysis and visualizations.

LabelingSample.ipynb: This file contains all code related to creating the sample for labeling, splitting the sample into batches to be given to domain experts and processing and aggregating the labeled batches.

7.6 *Data cleaning pseudocode*

Below is the pseudocode for the text cleaning pipeline performed on the exception user description text. The actual pipeline was implemented in Python code.

FUNCTION CLEAN_USERDESC(text):

1. Normalize input - Convert text to string, remove parentheses, hyphens/underscores → spaces - Remove role/position tags (OP, QA, Manager, etc.) - Replace "informed" → [NOTIFIED]

2. Replace structural patterns - Remove "vd/VD" tokens - Replace numbers → [NUMBER] - Replace alphanumeric IDs → [IDENTIFIER] -

Replace “form” or “sop” → [FORM/SOP] - Replace machine names → [MACHINE] - Normalize “entry error” variations

3. Split and standardize tokens - Break camelCase words - Remove single-letter tokens (A., B, etc.) - Remove leading punctuation - Fix spacing around punctuation and colons

4. Replace names with [PERSON] - Build STAFF_LIST from: * First + last name combinations * Single name tokens * Known misspellings/extra names - Match staff names using regex chunks - Replace each match with [PERSON] and record matched names - Collapse runs like “[PERSON], [PERSON] and [PERSON]” → single [PERSON]

5. Additional cleanup - Remove “ea”, "+", "/", bullet characters - Collapse punctuation runs - Collapse multiple spaces - Ensure one space before/after punctuation norms - Tidy remaining text

6. Spelling correction (SymSpell) - Preserve placeholders ([PERSON], [NOTIFIED], [FORM/SOP]) - Preserve domain acronyms and whitelisted terms (MES, PLC, etc.) - For remaining alphabetic tokens, apply SymSpell correction - Cache corrections for speed - Preserve original word casing patterns

7. Return results - Output: * Cleaned text string * Deduplicated list of names that were replaced by [PERSON]

END FUNCTION

7.7 *Note on BERTopic CountVectorizer*

As illustrated in Figure 3 in the Literature Review chapter, the CountVectorizer component of the BERTopic workflow does not affect topic formation, but only the words used to visually represent topics. Therefore, the CountVectorizer does not impact the clustering or label propagation aspects of BERTopic, nor does it affect the training and evaluating of downstream classifiers. However, a custom CountVectorizer was included in the BERTopic models only to make the resulting document-topic table outputs more easily readable for colleagues should they wish to review the unsupervised topics found by BERTopic outside of the label propagation exercise. The custom CountVectorizer function used Regex expressions to normalize placeholders inserted in the text during cleaning and used Spacy’s (Honnibal et al., 2020) lemmatizer function to strip words to their root stems.

REFERENCES

Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., & Hassan, A. (2023). Topic modeling algorithms and applications: A survey. *Information Systems*, *112*, 102131. https://doi.org/10.1016/j.is.2022.102131

AI, O. (2025). Text-embedding-large-3. Retrieved November 20, 2025, from https://platform.openai.com

Angelov, D. (2020, August). Top2Vec: Distributed Representations of Topics [arXiv:2008.09470 [cs]]. https://doi.org/10.48550/arXiv.2008.09470

Asas, V., Juan, S. S., Kerbun, V. C., Chua, S., Labadin, J., & Lau, E. (2025). A Comparative Analysis of Topic Modelling Techniques for Malaysian Business News Data: LDA, NMF, Top2Vec, and BERTopic. *2025 14th International Conference on Information Technology in Asia (CITA)*, 42–47. https://doi.org/10.1109/CITA66455.2025.11198734

Babikov, I., Kovalchuk, S., & Soldatov, I. (2023). Semi-supervised method for improving general-purpose and domain-specific textual corpora labels. *Procedia Computer Science*, *229*, 168–176. https://doi.org/10.1016/j.procs.2023.12.018

Berkson, J. (1944). Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association*, *39*(227), 357–365. https://doi.org/10.1080/01621459.1944.10500699

Bianchi, F., Terragni, S., & Hovy, D. (2021). Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 759–766. https://doi.org/10.18653/v1/2021.acl-short.96

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*, 106. https://doi.org/10.1186/1471-2105-14-106

Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of Science [arXiv:0708.3601 [stat]]. *The Annals of Applied Statistics*, *1*(1). https://doi.org/10.1214/07-AOAS114

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022. https://dl.acm.org/doi/10.5555/944919.944937

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Bu, W., Shu, H., Kang, F., Hu, Q., & Zhao, Y. (2023). Software Subclassification Based on BERTopic-BERT-BiLSTM Model [Publisher: Multidisciplinary Digital Publishing Institute]. *Electronics*, *12*(18), 3798. https://doi.org/10.3390/electronics12183798

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J., & Blei, D. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems*, *22*. Retrieved October 23, 2025, from https://proceedings.neurips.cc/paper_files/paper/2009/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html

Churchill, R., & Singh, L. (2022). The Evolution of Topic Modeling. *ACM Comput. Surv.*, *54*(10s), 215:1–215:35. https://doi.org/10.1145/3507900

Dieng, A. B., Ruiz, F. J. R., & Blei, D. M. (2019, October). The Dynamic Embedded Topic Model [arXiv:1907.05545 [cs]]. https://doi.org/10.48550/arXiv.1907.05545

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology*, *38*(1), 188–230. https://doi.org/10.1002/aris.1440380105

Egger, R., & Yu, J. (2022). A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts [Publisher: Frontiers]. *Frontiers in Sociology*, *7*. https://doi.org/10.3389/fsoc.2022.886498

Gan, L., Yang, T., Huang, Y., Yang, B., Luo, Y. Y., Richard, L. W. C., & Guo, D. (2024). Experimental Comparison of Three Topic Modeling Methods with LDA, Top2Vec and BERTopic. In H. Lu & J. Cai (Eds.), *Artificial Intelligence and Robotics* (pp. 376–391). Springer Nature. https://doi.org/10.1007/978-981-99-9109-9_37

Garbe, W. (2025). SymSpell [original-date: 2014-03-25T11:01:35Z]. Retrieved November 20, 2025, from https://github.com/wolfgarbe/SymSpell

Gottumukkala, D. P., P. V.g.d, P. R., & Rao, S. K. (2025). Topic modeling-based prediction of software defects and root cause using BERTopic, and multioutput classifier [Publisher: Nature Publishing Group]. *Scientific Reports*, *15*(1), 25428. https://doi.org/10.1038/s41598-025-11458-0

Grootendorst, M. (2022, March). BERTopic: Neural topic modeling with a class-based TF-IDF procedure [arXiv:2203.05794 [cs]]. https://doi.org/10.48550/arXiv.2203.05794

Hankar, M., Kasri, M., & Beni-Hssane, A. (2025). A comprehensive overview of topic modeling: Techniques, applications and challenges. *Neurocomputing*, *628*, 129638. https://doi.org/10.1016/j.neucom.2025.129638

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*.

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). Spacy: Industrial-strength natural language processing in python. https://doi.org/10.5281/zenodo.1212303

Huseynova, K., & Isbarov, J. (2024, August). Enhanced document retrieval with topic embeddings [arXiv:2408.10435 [cs]]. https://doi.org/10.48550/arXiv.2408.10435

Kazanci, N. (2025). Extended topic classification utilizing LDA and BERTopic: A call center case study on robot agents and human agents. *Applied Intelligence*, *55*(5), 360. https://doi.org/10.1007/s10489-024-06106-5

Kim, J., & Lee, S. (2024). Technology Opportunity Analysis for Creating Innovative Solutions: Applying Semi-supervised Topic Modelling on Patent Data [ISSN: 2159-5100]. *2024 Portland International Conference on Management of Engineering and Technology (PICMET)*, 1–9. https://doi.org/10.23919/PICMET64035.2024.10653159

Laureate, C. D. P., Buntine, W., & Linger, H. (2023). A systematic review of the use of topic models for short text social media analysis. *Artificial Intelligence Review*, *56*(12), 14223–14255. https://doi.org/10.1007/s10462-023-10471-x

Li, C., Duan, Y., Wang, H., Zhang, Z., Sun, A., & ma, Z. (2017). Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems*, *36*, 1–30. https://doi.org/10.1145/3091108

Li, X., Zhang, A., Li, C., Guo, L., Wang, W., & Ouyang, J. (2019). Relational Biterm Topic Model: Short-Text Topic Modeling using Word Embeddings. *The Computer Journal*, *62*(3), 359–372. https://doi.org/10.1093/comjnl/bxy037

Li, Y., Yang, Y., Song, P., Duan, L., & Ren, R. (2025). An improved SMOTE algorithm for enhanced imbalanced data classification by expanding sample generation space [Publisher: Nature Publishing Group]. *Scientific Reports*, *15*(1), 23521. https://doi.org/10.1038/s41598-025-09506-w

Ma, L., Chen, R., Ge, W., Rogers, P., Lyn-Cook, B., Hong, H., Tong, W., Wu, N., & Zou, W. (2025). AI-powered topic modeling: Comparing LDA and BERTopic in analyzing opioid-related cardiovascular risks in women. *Experimental Biology and Medicine*, *250*, 10389. https://doi.org/10.3389/ebm.2025.10389

McInnes, L., Healy, J., & Astels, S. (2017). Hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205. https://doi.org/10.21105/joss.00205

McInnes, L., Healy, J., & Melville, J. (2020, September). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

[arXiv:1802.03426 [stat]]. https://doi.org/10.48550/arXiv.1802.03426

Mekala, D., & Shang, J. (2020). Contextualized Weak Supervision for Text Classification. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 323–333. https://doi.org/10.18653/v1/2020.acl-main.30

Miao, Y., Yu, L., & Blunsom, P. (2016, June). Neural Variational Inference for Text Processing [arXiv:1511.06038 [cs]]. https://doi.org/10.48550/arXiv.1511.06038

Mishra, M. (2024). A holistic review of customer experience research: Topic modelling using BERTopic. *Marketing Intelligence & Planning*, *43*(4), 802–820. https://doi.org/10.1108/MIP-09-2023-0457

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic Evaluation of Topic Coherence. Retrieved October 23, 2025, from https://www.semanticscholar.org/paper/Automatic-Evaluation-of-Topic-Coherence-Newman-Lau/8e31f3c7e70e9a5f8afafd86cebc004d5eca8c2b

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*(2), 111–126. https://doi.org/10.1002/env.3170050203

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://arxiv.org/abs/1201.0490

Qiang, J., Qian, Z., Li, Y., Yuan, Y., & Wu, X. (2022). Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1427–1445. https://doi.org/10.1109/TKDE.2020.2992485

Ravenda, F., Bahrainian, S. A., Raballo, A., Mira, A., & Crestani, F. (2025). A self-supervised seed-driven approach to topic modelling and clustering. *Journal of Intelligent Information Systems*, *63*(1), 333–353. https://doi.org/10.1007/s10844-024-00891-8

Rezaei, M. R., Hafezi, M., Satpathy, A., Hodge, L., & Pourjafari, E. (2024, October). AT-RAG: An Adaptive RAG Model Enhancing Query Efficiency with Topic Filtering and Iterative Reasoning [arXiv:2410.12886 [cs]]. https://doi.org/10.48550/arXiv.2410.12886

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 399–408. https://doi.org/10.1145/2684822.2685324

Rüdiger, M., Antons, D., Joshi, A. M., & Salge, T.-O. (2022). Topic modeling revisited: New evidence on algorithm performance and quality metrics [Publisher: Public Library of Science]. *PLOS ONE*, *17*(4), e0266325. https://doi.org/10.1371/journal.pone.0266325

Sabbagh, R., & Ameri, F. (2019). A Framework Based on K-Means Clustering and Topic Modeling for Analyzing Unstructured Manufacturing Capability Data. *Journal of Computing and Information Science in Engineering*, *20*(011005). https://doi.org/10.1115/1.4044506

Sala, R., Francalanza, E., & Arena, S. (2025). A review on three decades of manufacturing maintenance research: Past, present and future directions [Publisher: Taylor & Francis]. *Production & Manufacturing Research*, *13*(1), 2469037. https://doi.org/10.1080/21693277.2025.2469037

Söderwall, A., & Telešova, G. (2025). A Comparative Study with LDA and BERTopic: AI Policies Across Different Democracy Indexes [Accepted: 2025-10-06T13:34:43Z]. Retrieved October 21, 2025, from https://gupea.ub.gu.se/handle/2077/89821

Srivastava, A., & Sutton, C. (2017). AUTOENCODING VARIATIONAL INFERENCE FOR TOPIC MODELS. *ICLR 2017*. https://arxiv.org/abs/1703.01488

Vayansky, I., & Kumar, S. A. P. (2020). A review of topic modeling methods. *Information Systems*, *94*, 101582. https://doi.org/10.1016/j.is.2020.101582

Wang, J., & Hsu, C.-C. (2020). A topic-based patent analytics approach for exploring technological trends in smart manufacturing. *Journal of Manufacturing Technology Management*, *32*(1), 110–135. https://doi.org/10.1108/JMTM-03-2020-0106

Wang, Y., Benavides, R., Diatchenko, L., Grant, A. V., & Li, Y. (2022). A graph-embedded topic model enables characterization of diverse pain phenotypes among UK biobank individuals. *iScience*, *25*(6), 104390. https://doi.org/10.1016/j.isci.2022.104390

Wankmüller, S. (2023). A comparison of approaches for imbalanced classification problems in the context of retrieving relevant documents for an analysis. *Journal of Computational Social Science*, *6*(1), 91–163. https://doi.org/10.1007/s42001-022-00191-7

Xiong, H., Cheng, Y., Zhao, W., & Liu, J. (2019). Analyzing scientific research topics in manufacturing field using a topic model. *Computers & Industrial Engineering*, *135*, 333–347. https://doi.org/10.1016/j.cie.2019.06.010

Xu, S., Wang, Y., Cheng, X., & Yang, Q. (2025). Thematic Identification Analysis of Equipment Quality Problems Based on the BERTopic

Model [ISSN: 2352-5428], 484–491. https://doi.org/10.2991/978-94-6463-676-5_47

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. *Proceedings of the 22nd international conference on World Wide Web*, 1445–1456. https://doi.org/10.1145/2488388.2488514

Zhang, Z.-W., Jing, X.-Y., & Wang, T.-J. (2017). Label propagation based semi-supervised learning for software defect prediction. *Automated Software Engineering*, *24*(1), 47–69. https://doi.org/10.1007/s10515-016-0194-x

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Scholkopf, B. (2004). Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems*. Retrieved November 5, 2025, from https://dl.acm.org/doi/10.5555/2981345.2981386

Zhou, Y., Lin, H., Liu, Y., & Ding, W. (2019). A novel method to identify emerging technologies using a semi-supervised topic clustering model: A case of 3D printing industry. *Scientometrics*, *120*(1), 167–185. https://doi.org/10.1007/s11192-019-03126-8