# Realization of Inter-Model Connections: Linking Requirements and Computer-Aided Design

by

Cheng Chen

(Under the Direction of Beshoy Morkos)

## Abstract

Managing rapid engineering changes in requirements and complex computer-aided design (CAD) models continue to increase the risk of industrial project failures in smart manufacturing. As products evolve over time, tracking design changes across different domains has become increasingly difficult to operate. Mismanagement incidents can derail industrial product development and result in financial losses. Existing practices often lack connections to cross-domain analysis and rely on domain experts to interpret engineering change propagation. To reduce the burden of this taxing process, this study proposes computational tools as digital threads that assist engineers in understanding the correlations of change propagation. The proposed framework investigates three components of analyzing engineering changes within and across domains. Particularly, the work pertains to (1) a topic modeling approach to narrow down engineering changes within requirements topics, (2) a framework for recognizing mechanical designs based on point clouds representations, and (3) an approach to incorporating joint embedding to learn the correlation between requirements and CAD images. The study makes use of several datasets, including three different heterogeneous industrial requirements documents, ShapeNetCore, and synthetic image datasets. Using this framework, engineers can generate interpretable results and determine the correlations of text-to-text and text-to-images for complex systems. The outcome of this study can contribute to building digital threads and assisting designers to make informed engineering decisions, track change propagation within and across domains, and reduce unanticipated engineering changes.

Index words: [SMART MANUFACTURING, DIGITAL THREAD, REQUIREMENT MANAGEMENT, COMPUTER-AIDED DESIGN, JOINT EMBEDDING, TOPIC MODELING, POINT CLOUD]

# Realization of Inter-Model Connections: Linking Requirements and Computer-Aided Design

by

Cheng Chen

B.S., Century College of BUPT, Beijing, China, 2012

M.Sc., Florida Institute of Technology, 2016

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Doctor of Philosophy

Athens, Georgia

2022

# Realization of Inter-Model Connections: Linking Requirements and Computer-Aided Design

by

Cheng Chen

| | |
|---|---|
| Major Professor: | Beshoy Morkos |
| Committee: | Ramana M Pidaparti |
| | Shannon Quinn |
| | Jidong Yang |

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
August 2022

# Dedication

THIS PAGE IS OPTIONAL

# Acknowledgments

THIS PAGE IS OPTIONAL

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Solving engineering's grand challenges and achieving sustainable development goals requires a broad spectrum of fundamental knowledge and interdisciplinary understanding to develop innovative solutions for industry 4.0. The engineering skills and knowledge necessary to succeed in the manufacturing job market of the future will continuously evolve as new knowledge is continually generated. The development of digital threads has emerged as a compelling topic for industries and researchers to represent information flow within complex systems. Thus, it has become more critical than ever before to transform today's design and manufacturing systems from manufacturing physical products to relying on digital threads, which has led to this study.

## 1.1 Motivation

Customer requirements management determines the success of today's industrial projects. Mismanaging requirements can cause more than half of the project failures and financial lost (PMI, 2014). Requirements management is a formal design process starting with customers' needs (CNs) and outputs a structured engineering design document. At every stage of the product lifecycle management process, requirements management plays a crucial role in addressing, adjusting, and verifying stakeholder expectations.

## Requirement Management

Requirement management is a process of assisting designers in documenting, analyzing, and tracking information throughout all product lifecycle stages. As requirements play a vital role in product evolution, organizations and industries across all fields experienced project failures and wasted program dollars due to insufficient resource allocation for requirement management (PMI, 2014). Moreover, unexpected engineering changes also contribute to project management failures (B. Morkos et al., 2012). Researchers have a broad understanding of requirements management and the type of representations and reasoning involved. In this chapter, requirements management refers to utilization of numerical models to identify and analyze similar design criteria based on latent topics. Manual entry into requirements management tools and the traceability of requirement changes still need to be improved for current industrial practice (Kropsu-Vehkapera et al., 2009).

The ability to establish requirements correlations and track engineering changes throughout the product life cycle is necessary to the success of a complex design. As engineering design and manufacturing systems become increasingly sophisticated, more unstructured requirements documents are collected through the iterative design process (L. Wang et al., 2021). Unstructured requirements [1] often contain domain-specific knowledge and concise information (C. Chen et al., 2021), making it difficult to

[1] Unstructured requirements are presented as text-heavy natural language data in the form of MS Word documents or Excel spreadsheets. It can be challenging for system engineers to handle ambiguous requirements when analyzing unstructured textual data.

understand and manage engineering changes. Extracting and analyzing useful design information from large unstructured data requires extensive manual intervention and communication, resulting in a human task that is taxing and prone to errors (Hein et al., 2018; B. Morkos et al., 2012). To address these challenges, this study examines several topic modeling approaches for generalizing requirements documents into topics that will assist engineers in understanding the structure of complex system with model-based systems engineering.

## Model-Based Engineering

As artificial intelligence advances manufacturing corporations, this evolution redefines both industrial business model innovation and reforms the manufacturing sector by introducing more data-driven decision making for each step of the manufacturing process. One of the most promising approaches, model-based enterprise (MBE)[1] , has shown its potential to drive smart manufacturing by linking all sources of digital data through the product lifecycle (Lubell et al., 2012). The global net value of the MBE market has grown from $7.89 billion in 2017 ("Model Based Enterprise Report 2019 - Global Market Outlook 2017-2026 - ResearchAndMarkets.com", 2019) to $9.94 billion in 2019 (*Global Model Based Enterprise Market - Industry Analysis and Forecast (2020-2027) - By Deployment Type, Offering, Industry and Region.* 2020), and the forecast for the future market performance is set at about $44 billion by 2027. Beyond upgrading the manufacturing equipment, companies have sought a digital model-based network for higher production efficiency and a profitable return on investment. Through machine learning techniques, building the next generation of manufacturing

[2] Model-Based Enterprise and Model-Based Engineering are indistinguishable terms. For clarity, MBE is defined as follows (Lubell et al., 2012):

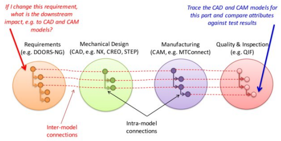
- Model-Based Enterprise refers to an organization that uses model-based engineering.
- Model-Based Engineering is a strategy for product development, manufacturing, and lifecycle while using a network approach (e.g., digital threads) to connect engineering activity.

networks will provide seamless product record-tracking and tracing capabilities to all parties, from customers to government regulatory compliance agents (Bajaj and Hedberg Jr, 2018; Davis et al., 2012).

The advances and implementation of MBE in engineering enterprises present an opportunity to understand an area of design practice that has been rarely explored. The MBE presents a unique opportunity to link all digital data sources throughout the product lifecycle, allowing data to be more adaptable to change even as manufacturing productivity increases to levels previously unthinkable. Developing such a system would allow for the tracking of design changes both upstream and downstream in requirements analysis and computer-aided design (CAD). For instance, changes from requirements can be subsequently realized in the CAD domain, and vice versa. Further, consider how requirements in CAD may be realized in the requirement domain. This is particularly important as requirements often serve as the contractual agreement between parties, and thus all changes and decisions need to align with that of the requirements. However, this is difficult to perform as relationships between requirements and CAD are not formalized nor fully realized. Often, it is dependent on experts to manually determine the relationship. If this was automated, engineers and designers could make informed decisions regarding requirements and CAD.

Note that the scope of this paper is set to develop the framework to perform a future system within MBE. In this paper, we therefore present a framework for research that will explore the links between requirements and CAD. In the existence of a multilevel information framework of MBE, digital threads [3] can be developed to synchronize data throughout the entire product lifecycle (T. D. Hedberg et al., 2020). During the conceptual design phase (e.g.,

[3] The concept of digital threads, as shown below, includes link-data methods and standard-based approaches that allow heterogeneous data from a variety of phases and systems to be compared, synchronized, and repaired across the entire product lifecycle.


If I change this requirement, what is the downstream impact, e.g. to CAD and CAM models?
Trace the CAD and CAM models for this part and compare attributes against test results
Requirements (e.g. DOORS-NG)
Mechanical Design (CAD, e.g. NX, CREO, STEP)
Manufacturing (CAM, e.g. MTConnect)
Quality & Inspection (e.g. QIF)
Inter-model connections
Intra-model connections

(T. D. Hedberg et al., 2020)

requirement management), design information, such as requirement changes, can be classified into four categories and visualization can be performed to determine the different change patterns over time (Giffin et al., 2009) with the likelihood of change propagations (Clarkson et al., 2004). Researchers can further predict the higher-order change propagations for a complex system (B. Morkos et al., 2012). Further, requirements can be also analyzed by lexical, syntactic, and structure analysis, and this approach has the potential to connect with CAD (Z. Y. Chen et al., 2007). For the mechanical modeling (e.g., CAD), most of the research focused on the applications related to graphics, analysis of components, computer numerical control, and manufacturing processes (Groover and Zimmers, 1983). Prior to this study, little research has been able to establish the correlation between requirements and CAD models. By utilizing machine learning techniques, engineering changes within requirements and CAD analysis can be performed coherently.

Advances in smart factories, coupled with the disruptions of supply chains, have created a turning point in manufacturing industries. With the increasing application of machine learning in design automation, model-based engineering (MBE) has become the new norm for handling manufacturing data. Despite the improvement in manufacturing resilience, we have not fully exploited semi-structured or unstructured data for design improvement. How to process multi-source data to aid knowledge acquisition during the design process has received attention in recent years from other industry environments, such as the process industry (Mao et al., 2019), the manufacturing execution system (Y. Wang et al., 2018), and the cyber-physical system (Cheng et al., 2018). In response to this information gap, designing a complex system would require the development of new tools and processes

(Castet, 2017). This means that domain experts should actively develop various design techniques to resolve dynamic engineering change management issues.

## Mechanical Designs

Using computer vision to recognize different objects and shapes has become increasingly important in the field of manufacturing (Lyu et al., 2021), autonomous driving (Kidono et al., 2011), and augmented reality (Alexiou et al., 2017). To overcome certain technological limitations, many industries have shifted from using 2D images to capturing 3D geospatial data. As more data is being collected by various types of sensors, such as LiDAR, the challenge of recognizing objects from point clouds has gained more attention in recent years. In addition, the classification of targeted objects in a real-world environment would require a more robust and computationally efficient model (Uy et al., 2019). In a manufacturing environment, segmenting a point cloud into mechanical components or subassemblies can assist designers in identifying objects as well as in detecting potential product defects in advanced manufacturing. However, few approaches have implemented point clouds into design manufacturing applications due to the limited availability of benchmark datasets and the lack of algorithmic development. As a result of the implementation of point clouds in design and manufacturing, computer vision systems are becoming increasingly capable of recognizing mechanical designs, geometric characteristics, and mechanical subassemblies automatically. A more robust design tool will allow engineers to make better decisions and achieve lean manufacturing by aiding engineering changes.

## 1.2 Challenges in Model-Based Enterprise

Analyzing data from a variety of sources presents its own set of challenges. First, due to confidentiality, few design documents are publicly available or can be used for benchmark datasets. Second, extracting meaningful information from unstructured datasets is difficult. Unstructured data in engineering design often takes the form of textual information, such as design discourse (Gyory et al., 2020) or customer feedback (Song et al., 2020). Many natural language processing techniques are often applied to retrieve useful information from domain-specific data. In requirement management (RM), the corresponding image datasets are rarely documented. To make up for the missing information, image scraping is used to collect online images based on the given requirements. With the combination of textual and visual information, our study presents a framework for bridging the information gap between unstructured requirements and synthetic image datasets.

Despite the potential advantages of promising technologies (e.g., MBE), some barriers may hinder the transition (Nathan Hartman, 2018). One of the challenges is to add a decision support layer in a local supply chain network (Davis et al., 2012). Managing an entire information system requires a more efficient business and operating model, which enables the model-based system to manage automation, optimization, and decision-making across different manufacturing infrastructures. Second, every organization employs various product lifecycle management (PLM) tools/software to build a fully designed model, and few companies can afford such integrated software shared with their suppliers. In response, MBE software (e.g., Syndeia[2]) integrates different domain platforms with various standard-based data, using digital threads.

The goals of Model-Based Engineering are data repair, synchronization, and sharing; digital threads connect the information flow among all phases of the product lifecycle (T. D. Hedberg et al., 2020). Furthermore, many leaders of major manufacturing sectors accept the MBE concept and envision that MBE can reduce the cost of the technology management process by 50% and reduce time to market by 45% (Bajaj et al., 2016; T. Hedberg et al., 2016). In PLM, data management still lacks detailed techniques and formal studies to support decision-making (J. Li et al., 2015). Therefore, this paper proposes a fundamental framework for understanding the relationship between requirement management and CAD modeling.

## Research Challenges of Generating Digital Threads

Every design journey begins with requirements eliciting, analyzing, and specifying design information to satisfy the needs of stakeholders. Especially for complex systems, the large amount of engineering design documentation collected and generated for a product can make it difficult to navigate and retrieve specific correlations among the requirements (Saaksvuori and Immonen, 2008). For creating digital threads, many NLP tools are employed to tackle the challenging issues within design documents (Ball and Lewis, 2020; Gyory et al., 2021; Joung and Kim, 2021; Saidani et al., 2021); however, requirements documents present their own challenges when it comes to obtaining information such as:

- Discovering, analyzing, and representing domain-specific design topics from a small collection of requirements documents
- The extraction of useful information from sparse and high-dimensional textual data

[4] http://intercax.com/products/syndeia/

Depending on the requirement style, both functional and nonfunctional requirements may contain various frequency distribution of domain-specific terms. Current methods, however, are not adequate to capture the semantic relationship among words with low lexical frequencies. As most requirements contain few words, explicitly implementing NLP models may achieve limited results. Therefore, the need for more robust models to manage requirements in industrial applications has become increasingly important.

### Challenges in Multi-source Data

Exploiting collected data from multiple sources is a challenging task that must be accomplished to satisfy the requirements and ensure the success of industrial projects. Various industries may utilize different formats or standards for production and design. Often, manufacturers are unaware of how to obtain and store design information (J. Li et al., 2015). Among the stored information, many data-driven tools require the availability of large and structured datasets. It is still difficult to fully exploit unstructured or semi-structured data, including text, images, and video. To integrate unstructured data into current PLM systems, more data-driven approaches should be developed to provide cost-effective solutions.

### Data Challenges in PLM

Several major challenges of implementing PLM (e.g., product design, manufacturing, and customer service) in manufacturing sectors remain. The PLM front-end is a centralized network where vendors manage all product information. Although data collection has grown rapidly, the “Big Data” concept and technique still have limited application in the PLM domain (J. Li et al., 2015). Due to the difficulty of integrating, sharing, and storing distinct types of data, current solutions rely on software for managing, analyzing, and simulating data. For instance, image files are often included with technical notes in a folder tree sent to suppliers (David and Rowe, 2016). Often, requirements exist as interface layers in PLM software, and it may be difficult to provide designers with direct visualization.

## 1.3 Research Objectives and Deliverables

The motivation for this research can be categorized into four components, as shown in Table 1.1. Each research objective is accompanied by a high-level summary of the outcomes. Deliverables contain existing and ongoing papers in the form of conference or journal proceedings.

The proposed study consists of two major stages: a requirement management study and a CAD study. We aim to answer two research questions for a given product: how to categorize requirement data into topics and how to associate geometric models (e.g., CAD) with requirement domains. First, an in-depth study is presented in Figure 1.1, which illustrates the combined knowledge representation between requirements and CAD, where digital threads correlate different design information throughout all phases of the product lifecycle. For instance, the requirements might not necessarily describe all the design details for CAD, and CAD component designs cannot directly translate the design specifications back to requirements. Accordingly, if requirement sentences and CAD data can be learned jointly, we hypothesize that by converting natural language requirements into subspaces, we can categorize requirements based on their semantic structures. Second, current literature lacks a descriptive method to define the connectivity among CAD

| Objectives | Outcomes | Deliverable | Dissertation |
|---|---|---|---|
| Investigate on how to implement topic modeling on requirements documents | Estimate the appropriate number of topics | Conference Paper (C. Chen et al., 2021) | Chapter 3, Section 3.2 |
| Explore different types of NLP model combinations for representing requirements documents | Analyze the optimal combination of models for each industrial requirements documents | Journal Paper | Chapter 3, Section 3.3 |
| Design a framework to cluster CAD components into mechanical subassembly | Develop and evaluate a proposed neural network model to classify point cloud data into categories | Conference Paper (Mohammadi et al., 2022) | Chapter 3, Section 3.4 |
| Implement a joint embedding model to learn the correlation between requirements and mechanical images | Compare and investigate the improved performance for a fine-tune model | Journal Paper | Chapter 3, Section 3.5 |

Table 1.1: Research Objectives, Outcome, and Deliverables

components. However, by learning a joint representation, CAD parts can be partitioned into separate groups while maintaining functional reasoning from requirements.

## 1.4 Solution Overview

Many studies have modeled engineering changes using matrix-based modeling as a general approach, such as the design structure matrix (DSM)[5] (Browning, 2015; Eckert et al., 2004; Hein et al., 2018; Lee and Hong, 2017; B. Morkos et al., 2012; Tilstra et al., 2012), graph theory, and system

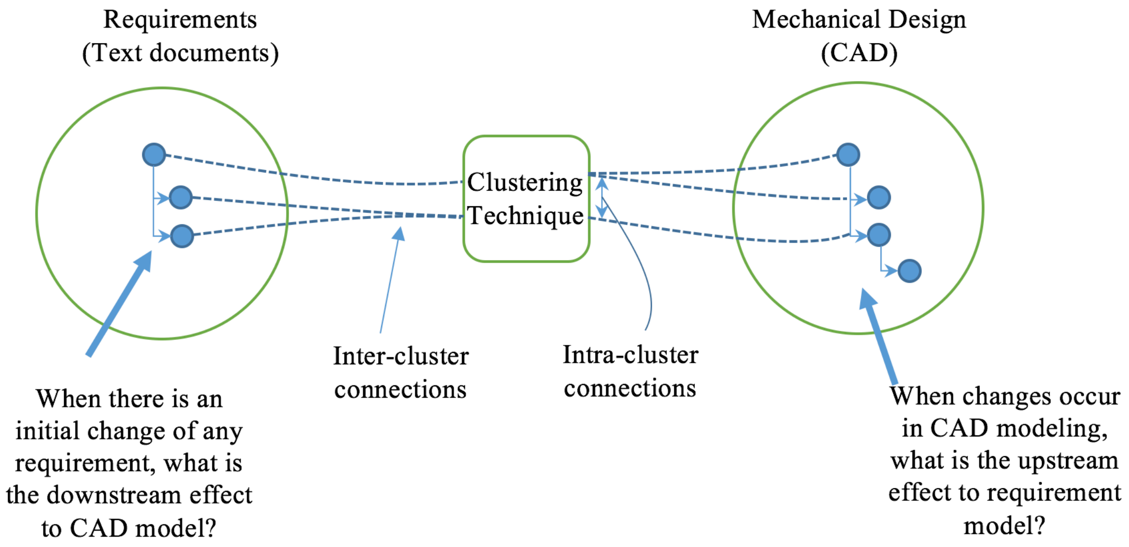


Figure 1.1: A Flow Chart Of Coding Process To Build Digital Threads For MBE

modeling language (SysML), to manage the interrelation of complex system requirements. When managing a complex system, design practitioners can examine DSMs to track engineering changes and locate the related requirements. The value of dependency relations depends on either rating schemes (Browning, 2001; Helmer et al., 2010), keywords (Mocko et al., 2007), scoring metrics (Yu et al., 2007), or attributes (Y. Chen et al., 2010). However, converting requirements documents to a matrix representation requires domain experts to interpret and maintain requirement changes. In contrast, graph representations of complex systems are often used to demonstrate how engineering changes affect their physical systems (Eckert et al., 2004). Both direct and indirect graphs can analyze the likelihood of change propagation and its downstream impact (Clarkson et al., 2004; Hein et al., 2021; Keller et al., 2009).

For process-oriented applications, SysML simulates operations and generates graph-based representations to trace engineering changes across domains. Among the most common tools used in industry projects are Astah,

[5] A square matrix,

| | Element 1 | Element 2 | Element 3 | Element 4 | Element 5 | Element 6 |
|---|---|---|---|---|---|---|
| Element 1 | 1 | ● | | ● | ● | ● |
| Element 2 | | 2 | ● | ● | | |
| Element 3 | | | 3 | | ● | |
| Element 4 | | | | 4 | ● | ● |
| Element 5 | | | | | 5 | ● |
| Element 6 | | ● | | | ● | 6 |

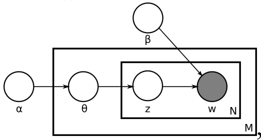
, in which the off-diagonal elements represent the dependencies between elements. Diagonal elements don't contribute significant meaning (Browning, 2015).

IBM Rational DOORS, NoMagic MagicDraw, IBM requirement quality assistant (RQA), and Jama Connect (B. W. Morkos, 2012). Using these tools, engineers can organize logical relationships among requirements, share interpretable data among teams, and specify the capabilities of a system. Such a process depends on intensive human efforts and specific domain expertise at each design stage. Therefore, a more automated process is preferable for developing a probabilistic model framework and understanding the requirement correlations, which will provide additional relevant information for designers to make informed decisions. Design practitioners can trace requirement changes based on each subsystem and narrow down the potential change paths.

## 1.5 Proposed Methods

One feasible way to improve requirements management is to use topic modeling. Topic models are a type of statistical model used in natural language processing (NLP) capable of discovering interpretable "topics" for textual data. One prominent technique, latent Dirichlet allocation (LDA) [6] (Blei et al., 2003) is studied extensively. LDA is a hierarchal Bayesian model for revealing the latent semantic structure of documents within a corpus based on their semantics. The LDA approach, in contrast to other approaches, such as word embedding, assumes that each corpus contains a mixture of topics and that the order is irrelevant. Each document is assumed to be a collection of topics, and each word contributes to multiple topics with varying probabilities. These assumptions are also valid for requirements documents, where documents can be randomly shuffled and then divided into both training and hold-out sets. However, limited research is conducted

[6] A diagram of LDA's model architecture is shown below



, where topic-word and document-word probability distributions are computed as part of the training process (Commons, 2020).

on applying LDA in requirements documents, though this process could be problematic for generating differentiable and interpretable results. Other models, such as word embedding with clustering, hierarchical Dirichlet process (HDP), short text topic modeling (STTM), Gibbs sampling algorithm for the Dirichlet multinomial mixture model (GSDMM), and hierarchical latent Dirichlet allocation (hLDA), can also accomplish comparable results with various assumptions. To avoid the need to make further assumptions for design requirements, we have chosen LDA as a baseline model to study design requirements. Thus, the scope of this study is to implement LDA to requirements documents and determine topical representations for each dataset.

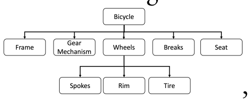
Within many existing techniques for performing semantic analysis, LDA is widely used to extract latent topics from a collection of documents (Blei, 2012). The LDA is a generative probabilistic model that creates topics based on large observations and predicts the topic composition of unobserved documents. The number of interpretable topics is typically predetermined by domain experts. However, LDA has limited performance for generalizing topics from short documents with few words, such as tweets and Reddit posts. For short sentences, several variations of LDA have been developed, such as short text clustering (Qiang et al., 2020; Yin and Wang, 2014). For instance, the movie group process analogy provides an insightful understanding of the GSDMM, in which students can be clustered into $K$ tables based upon their common interests in movies. As each movie title can appear only once, the clustered results strike a balance between completeness and homogeneity. As a result of different model assumptions, both models can generalize words into topics with varying levels of performance.

The emerging concept of identifying latent modules or subassemblies is studied as an intermediate step for organizing changes within subsystems of engineering products using unstructured data. To capture hidden topics or semantic representations within design documents, a variety of approaches in natural language processing can be utilized. As engineering changes rarely occur alone, identifying changes in information within modules could aid in narrowing the range of affected physical components. Several early studies integrated DSM and clustering algorithms to determine the modularity of product architectures (Jung and Simpson, 2017; Yu et al., 2007). On the other hand, other approaches have attempted to capture latent modules from text documents. Early studies have employed a variety of techniques, including term frequency-inverse document frequency (TF-IDF), latent semantic analysis (LSA), non-negative matrix factorization (NMF), and probabilistic latent semantic indexing (pLSI) (Ball and Lewis, 2019). To overcome certain limitations of previously developed methods, latent Dirichlet allocation (LDA) allows for analysis of design team communications (Gyory et al., 2021) and product ecosystems (Zhou et al., 2020), mapping of authorship networks (Guo et al., 2018), and filtering of key words (Joung and Kim, 2021). As opposed to the previous approaches, LDA can generalize corpora based on a predetermined number of word groups without relying on handcrafted correlations. As the original LDA model fails to capture hidden semantic similarity in short corpora, this study hypothesizes that the combination of topic models with word embedding algorithms will improve overall performance. Among the most popular word embedding models, sentence embeddings using Siamese BERT-networks (sentence-BERT) is a transformer-based neural network (Reimers and Gurevych, 2019), which is commonly pre-trained on large-scale corpora for the purpose of learning general language representations. Combining embedded knowledge from sentence-BERT with topic modeling could provide a more robust representation of requirements documents.

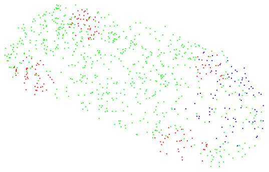
The different combinations of LDA and sentence-BERT will produce interpretable correlations and visualizations between topics and words. Using requirement topics as building blocks, engineers can further track design changes using topic-word correlations, generalize the structure of requirements documents, and predict the impact of forthcoming design changes without collecting extra information. As a measure of the model's performance, metrics (coherence scores and silhouette scores) and human judgment are used against three industry projects. The findings of this study can provide valuable insights into tracking the propagation of engineering changes in complex designs.

A product decomposition in mechanical design can be divided into modularity of the product and structural decomposition (Kusiak and Larson, 1995). Compared to product modularity, structural decomposition focuses on the breakdown of mechanical structures into parts and subsystems. Subassemblies represent subsystems [7] that may consist of several components. Few studies are conducted using data-driven approaches to identifying subassemblies in product architectures. To create an intelligent model for the automatic recognition of mechanical sub-assemblies for distinct designs, we need to identify the various categories of objects first. In this study, a different framework for recognizing geometric models is proposed using point cloud representations [8].

[7] A product subassembly, as shown in Figure below , is often represented as a tree diagram, where the wheels are the leaf nodes and have three children (Kusiak and Larson, 1995. Dependencies are indicated by the edges.



[8] Using an airplane as an example in the Figure below, the point cloud can accurately represent geometric models from ShapeNetCore datasets. Subsystems are represented by colors.



A promising method for learning the correlations between requirements and geometric models is to implement joint embedding, which is a machine learning technique that captures the association between both types of datasets. Early studies in this area employed different approaches to analyze texts and images in relation to each other. The concept of correlating sub-images with keywords (nouns and adjectives) was applied to predict the labels of new images (Mori et al., 1999). Further, a multimodal Deep Boltzmann Machine (DBM) model was proposed to learn joint distributions over images and texts (Srivastava and Salakhutdinov, 2012). Convolutional networks have shown their ability to recognize correlations within and between images and words as further evidence of advancing vision models (Joulin et al., 2016). Different deep convolutional neural network architectures were created to accurately label images (Krizhevsky et al., 2012; G. Li et al., 2020). As unforeseen images may have unanticipated labels, an N-gram model approach was trained on unlabeled datasets and predicted the possible labels for a testing set (A. Li et al., 2017). A contrastive language-image pre-training (CLIP) model was recently developed to deal with out-of-distribution predictions by using zero-shot learning (Radford et al., 2021). For typical image and text classification problems, both training and test datasets are from the same distribution. In contrast, the CLIP model uses a dot product to learn the joint embedding space and perform zero-shot prediction on images with truly out-of-distribution samples. Several pre-trained CLIP models containing general knowledge can be further fine-tuned to learn domain-specific designs. Because of these factors, we selected the CLIP model to learn the correspondence between requirements documents and images.

This work proposes a method to address the current information gap of multi-source data issues within MBE. We present a model that can learn domain-specific knowledge by building correlations between images and texts. Harvesting a variety of unstructured data enables interpretable visualizations for engineering changes, as shown in Figure 1.2. By using this method, engineers can visualize the interconnections of subsystems and manage change propagations.

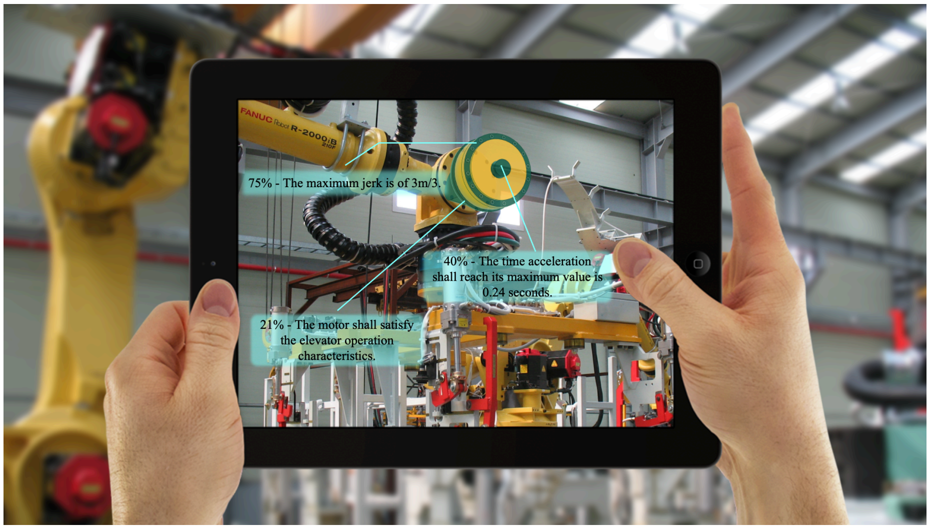

FANUC Robot R-2000iB 210F
75% - The maximum jerk is of 3m/3.
40% - The time acceleration shall reach its maximum value is 0.24 seconds.
21% - The motor shall satisfy the elevator operation characteristics.

Figure 1.2: A Conceptual Example Of The Use Of Multi-source Data In Manufacturing

## Propagating Engineering Changes Across Domains

A significant challenge during product lifecycle management is how to automatically interpret and translate engineering changes into domain-specific knowledge. One of the major obstacles is the absence of open-source datasets that can be used to study the impact of design changes made to CAD models. Little mechanical industrial design is available online and can be used as a benchmark dataset. It is common knowledge that larger available datasets would help to improve the performance of neural network models. In recent years, many popular datasets developed for computer vision have been used in numerous fields, including IMDB-Wiki Dataset, ShapeNetCore, ImageNet, Fashion MNIST, and CIFAR-10. Such datasets are gathered online and often annotated by humans to ensure quality. In mechanical design, large-scale datasets containing either 2D images or 3D CAD models associated with multiple types of design information are highly desirable.

## 1.6 Research Questions

Each generated topic from LDA consists of a list of words with corresponding probabilities, where designers can understand high-volume requirements documents through generated topics. Visualizing topics correlated with requirement sentences can reduce human error (Cerpa and Verner, 2009; Ullman, 1992) and improve design efficiency by organizing corpora based on topics. Further, analyzing latent topics can contribute to information tracking and developing digital threads across a product's lifecycle (T. D. Hedberg et al., 2020). Table 1.2 provides details on studying information tracking by examining three fundamental research questions (RQs):

The results of this study have implications for requirement management and the development of various phases of PLM. With the integration of our proposed framework into digital threads, the CLIP model can create and realize the connections between image and requirement data. By tracking these correspondences, designers can track engineering changes across various domains. Our findings could also impact engineering education for design practitioners on how to infuse domain-knowledge into AI to gain an in-depth understanding of complex systems. Further, this study could be expanded

| | |
|---|---|
| Research Question 1 | **RQ 1.1:** Can design requirements documents be interpreted based on the generated topics?<br>**Hypothesis:** Topic modeling may generate interpretable requirement topics that can be verified by domain experts.<br>**RQ 1.2:** How can we determine the number of topics generated or interpreted that adequately represent each requirement document?<br>**Hypothesis:** Depending on the perplexity and coherence values, the appropriate number of topics for each industrial project may be determined.<br>**RQ 1.3:** Can generated topics accurately represent the subsystems in each requirement corpus?<br>**Hypothesis:** By adjusting the relevance score, each requirement topic's quality can be improved. |
| Research Question 2 | **RQ 2.1:** How to design a computationally efficient model for differentiating a large database of 3D mechanical designs?<br>**Hypothesis:** The proposed model can detect various types of mechanical designs by incorporating meta-learning and SAE techniques.<br>**RQ 2.2:** How to improve prediction accuracy for the proposed model?<br>**Hypothesis:** During training, different types of random noise can be introduced into a point cloud dataset to achieve a greater level of generalization. |
| Research Question 3 | **RQ 3.1:** How to create a synthetic image dataset for representing the missing mechanical design information?<br>**Hypothesis:** A image retrieval technique can be used to locate the most relevant information.<br>**RQ 3.2:** How can transfer learning be used to establish correlations between requirements and images of mechanical components?<br>**Hypothesis:** Using a pre-trained foundation model can serve as a starting point for understanding domain-specific knowledge.<br>**RQ 3.3:** Can a fine-tuned model predict the most relevant sentences from domain-specific requirements documents?<br>**Hypothesis:** A zero-shot learning procedure can test the correlation between the most relevant image and the requirements. |

Table 1.2: Research Questions and Hypothesis

to combine multiple data sources and improve PLM's digital manufacturing capabilities at an early stage.

# Chapter 2

# Relevant Literature Review

To support the discussion in the remainder of this chapter, this chapter generalizes the necessary background knowledge for building digital threads under the framework of model-based engineering. First, a literature review regarding requirement management is reviewed and a research gap is identified to demonstrate the importance of introducing topic modeling in the development of digital threads. Second, topic modeling and word embedding techniques are discussed in detail to support the development of digital threads.

## 2.1 Requirements Management

Requirements play a critical role in the conceptual design phase, and they are often presented as a list of documents containing product design specifications/constraints (Hein et al., 2018; Pahl and Beitz, 2013). By consulting stakeholders, users, customers, or suppliers, requirements clarify design tasks and record the limitations for product development (Andreou et al., 2003; Fricker, 2010; Nilsson and Fagerström, 2006). For a complex system, testing and evaluating the complete requirements could prove challenging (Bloebaum and McGowan, 2012; Giffin et al., 2009). Moreover, the design is an iterative process, and any initial changes might result in an unanticipated change propagation due to different representations or insufficient communication among designers (Eckert et al., 2004; Kobayashi and Maekawa, 2001; Ncube and Maiden, 1999). To predict the most likely consequences, requirement propagation is defined based on their types and purposes (Z. Y. Chen et al., 2007; Z. Y. Chen and Zeng, 2006; Giffin et al., 2009). Much existing commercial software (e.g., IBM DOORS (Eriksson et al., 2005) or JAMA[1]) and many research tools (e.g., ARCPP (Hein et al., 2018), ROM Client (Z. Y. Chen et al., 2007)) can manage requirement repositories; however, their functionalities are incapable of representing the CAD models. To address this challenge, this study describes a scheme to cluster requirements as groups, using a spectral clustering method. If successful, this work would reduce the workload related to requirement documents and miscommunications among design engineering teams.

Document-based requirements are written in a specific format to avoid ambiguity and ensure testability for reflecting stakeholder needs. Requirements management consists of requirement elicitation, analysis, specification (Jiao and Chen, 2006), and verification. In requirements analysis, the goals include improving engineering processes such as requirements classification, prioritization, negotiation, or change propagation. One of the key issues in requirements analysis is the confirmation management (CM) topic evaluators list (Kapurch, 2010), where designers must verify and trace each design change manually. Visual analysis of requirements with topic modeling could assist in identifying and inspecting all appropriate changes.

### Functional and Non-Functional Requirements

Engineering changes happen at different levels of requirements, which can be generalized as functional requirements (FRs) and non-functional requirements (NFRs) (J. Summers and Morkos, 2013). A requirement hierarchy structure is often used to present the FRs as operational and technical requirements, where each technical requirement must trace back to an operational requirement (Cellucci, 2008). The concept of modeling requirements by topic can assist engineers in predetermining the range of requirements that may be affected during requirement management. This allows designers to visualize the topic composition and identify the relevant FRs or NFRs based on their domain knowledge.

**Current Study Progression** Based on the literature on requirements management, this study falls within the domain of requirements analysis. We initiated requirement management studies in our research group by managing various industrial projects (B. Morkos et al., 2010; B. Morkos et al., 2012). As a result of the initial study, correlations of requirements are often modeled using handcrafted features. To improve change traceability, an automated requirement change propagation prediction tool (ARCPP) is developed to track engineering changes within requirements documents automatically. As engineering changes are volatile in nature, the following study estimated information changes across nodes to measure requirement volatility (Hein, 2018). An advantage of having such a tool is the ability to measure which requirements are likely to lead to the most changes. This study, however, adopts a different approach by narrowing down the range of requirement changes using topic modeling (C. Chen et al., 2021). Rather than narrowing down the range of engineering changes manually, the designer could track changes based on pre-assigned topics. As a next step in this area of research, it is imperative to leverage transfer learning by implementing large foundation models for tracking and identifying engineering changes.

### Engineering Changes in Product Llifecycle Management

The engineering change process involves the creation, review, and approval of engineering change requests (ECRs) and engineering change orders (ECOs). Many research efforts have explored the development of tools for managing changes in reengineering processes using DSM-based methods. A literature review reveals the change prediction method (CPM) (B. Morkos et al., 2012) can capture requirements relationships using higher-order DSMs to track requirements changes and anticipate change propagation. Building on selected keywords, NLP techniques could predict engineering change propagations on vastly different design projects (B. Morkos et al., 2014). This finding has led to the current investigation of all words in textual information using topic modeling, to provide an alternative to the existing requirements tracing techniques that are based on DSMs.

The static DSM (e.g., affinity matrix, $A \in \mathbb{R}^{n \times n}$ ) represents the internal relationships among the requirements of a complex system for potential change propagations (Browning, 2001; B. W. Morkos, 2012). Each element of DSM defines a document or unique word. The off-diagonal component reveals the dependency of the pairwise comparison between any two subcomponents. Within DSM, various techniques can analyze and categorize requirements into subgroups/sub-diagonal blocks based on the concepts (words) (Danilovic and Browning, 2007; Y. Huang et al., 2012; Qiao et al., 2017; Yang et al., 2013). Since requirements can be generalized into a set of concepts between similar contexts, this study approaches requirement management with the aim to reduce the dimensionality of the dataset using a different clustering method.

Requirement management studies the areas of documenting, analyzing, and prioritizing technical design information. For developing a complex system, many existing approaches build the correlations among requirements to better understand and predict the propagation of information changes. However, the process of developing a topic layer to narrow down requirement change propagation has not been thoroughly studied. The purpose of this study is to improve the generation of design topics from requirements documents. This proposed framework explores different combinations of topic and word embedding models to determine which setting can extract the most relevant design information for design topics. To understand the reasoning behind this framework, the following section provides relevant background information and topic models on requirements management based on three industrial projects.

This study was prompted by earlier research on requirements management. As requirements documents are collaboratively developed based on different domain knowledge, tracking engineering changes within a domain can be problematic. Different techniques were employed to elicit requirements (B. Morkos and Summers, 2009). As the project progresses, the evolution of requirements significantly affected the success of the team (Joshi et al., 2019; B. Morkos et al., 2019; J. D. Summers et al., 2014). One of the most challenging aspects of requirement analysis is managing changes. There is evidence that requirements may not always correlate well with other populated design documents within a project (B. Morkos et al., 2010). Such discrepancies may

[9] https://www.jamasoftware.com

result in information loss during the propagation of engineering changes. An ARCPP tool has been developed to simulate the affected requirements based on keywords as each engineering change propagates through requirements (Hein et al., 2015; B. W. Morkos, 2012). Additionally, other methods such as centrality measures (Htet Hein et al., 2017) and neural networks (B. Morkos et al., 2014) were used to assess the properties of requirement networks and to compare prediction accuracy. In contrast to FRs, a case study demonstrated that engineering design decisions are often influenced by NFRs in the automotive OEM industry (Shankar et al., 2012). As a result of the high complexity of design change propagation, a volatility measure is designed to determine how engineering changes react in the following four predefined scenarios: multiplier, absorber, transmitter and robust (Hein et al., 2021). To reduce the risks caused by unexpected change propagation, the topic model approach generalized requirements documents into interpretable groups from which propagation can be estimated (C. Chen et al., 2021). However, the propagation of requirement changes should result in the realization of the physical components, and these connections are not well understood. For developing such correlations, this study examines various aspects of the RM process within PLM.

Distribution management is a process for approving engineering changes for documents within PLM (Saaksvuori and Immonen, 2008), where engineers spend 15-40 percent of their time searching and checking information within PLM systems. During the change verification process, requirements are used to ensure product design integrity and target performance. At every stage of the product lifecycle, the configuration of customized products requires making difficult trade-off decisions to comply with customer requirements.

These decisions are critical to the successful completion of complex projects (Giffin et al., 2009). In addition to decision making, the success of the product also depends on the allocation of appropriate resources for requirements management.

A requirement risk is a potential mismatch between stakeholder expectations and the outcome of a project. With the integration of RM tools into PLM (Violante et al., 2017), a product-centric approach becomes increasingly critical to trace design information related to the physical product. Such connections within PLM's subsystem can be classified as intra-model or inter-model connections (T. D. Hedberg et al., 2020). By enabling digital threads, engineering changes can be propagated across subsystems through these connections. In recognition that solving engineering changes alone can be viewed from many different perspectives, the leading practices can be divided into three categories: design teams (Terwiesch and Loch, 1999), computer-based tools (G. Huang and Mak, 1998), and model-based systems (Madni and Sievers, 2018). Different disciplines have different approaches to handling the challenges posed by passing design information to other domains. Recent merging problems have included how to learn and represent various types of data or how to improve user interface interoperability. Several perspectives on RM are presented in the following sections to minimize requirement risk.

## Existing Tools for Analyzing Requirements Changes

Often, companies adapt existing designs of products or machines to reduce the high costs and risks of developing new products while meeting customer needs (Cross, 2021). As part of the product development process, companies strive to bring the product to market as quickly as possible (Ulrich, 2003). Therefore, understanding the structure of existing documents and reusing such textual correlations can effectively assist engineers in redeveloping products and mitigating unexpected changes during the early stages of design.

The requirement management involves eliciting (B. Morkos and Summers, 2013), analyzing (Browning, 2015), specifying (Shankar et al., 2010), and verifying stakeholder needs. Requirement hierarchy and traceability are two major aspects of requirement management (Cellucci, 2008; Hirshorn, 2017). The requirement hierarchy indicates the level at which a set of requirements should be verified (e.g., from system to subsystem level or from operational to technical requirements). In confirmation management, requirement traceability refers to the ability to manage changes within the hierarchy of requirements throughout the entire life cycle of a product. The changes are frequently bidirectional, as engineering changes may be propagated either upstream or downstream. Automating the process of tracking requirements changes within a requirement hierarchy remains an open challenge.

### Latent Semantic Analysis

Early work often makes use of natural language techniques for investigating requirements changes within design documents. One of the most popular methods is Latent Semantic Analysis (LSA). LSA is a statistical technique to study the semantic and contextual reasoning of text documents (Deerwester et al., 1990; Foltz et al., 1998; Hofmann, 2013).

**Text Preprocess**. Text preprocess is an operation to transform every text into its canonical form. Since requirement documents contain many non-standard words, a standard preprocess is necessary for LSA to realize the digital contents. Lowercase, tokenization, lemmatization, and punctuation have been included in this preprocessing step using Python Spacy Package. Since some of the high-frequency words might still offer some values in representing the structure of requirements, only certain stop words have been eliminated under scrutiny. We also assume nouns, verbs, adverbs, and adjectives have equally important roles in capturing the connections among requirements; LSA analyzes those words all together. For instance, Table 3.6 shows the difference before and after this preprocessing.

Table 2.1 & Table 2.1: One Example Of Requirements From The Project 1

| **Original Requirement:** |
|---|
| 2.2. Each station shall be able to accommodate casing length of API Range Three from thirty four feet to forty eight feet. |
| **After Pre-process:** |
| station able accommodate case length api range three thirty four foot forty eight foot |

After the text preprocesses, the trimmed requirements have been used as inputs for LSA for further analysis. Typically, both bag-of-words (square matrix) and LSA (least square matrix) can represent the structure of requirement documents. However, the bag-of-words model is often employed as a sparse matrix with a high sparsity, and it can be computationally expensive since each word represents one dimension. For a large requirement document, the bag-of-words model has often encountered the curse of dimensionality issues. Instead, the TF-IDF model can calculate the weights for every single unique word (e.g., feature) corresponding to each document. Based on the TF-IDF scores, the final result shows the importance of each word for different documents and can describe the correlations among requirements with a low sparsity. Furthermore, N-gram models could also improve accuracy by including unique phrases for each concept. Since both of the projects contain mainly bigram terms (e.g., manual inspection, lifting mechanism), an n-gram range can potentially improve model performance for LSA.

LSA generates concepts based on correlations between a set of documents and their words. There are four major procedures, as follows (Fu et al., 2013; Landauer et al., 1998):

(1) For constructing a word-by-sentence matrix, each row (sentence) refers to one requirement sentence, and every column contains a unique word. A standard NLP preprocessing procedure, including tokenization, normalization, and feature extraction, can reduce the noise for the training set. Based on the occurrence of each unique word to each requirement sentence, a Term-Frequency (TF) records the total score within the word-by-sentence matrix.

(2) A Term Frequency - Inverse Document Frequency (TF-IDF) is a method to reduce the effect of high-frequency words in natural language (e.g., "a" and "the") (Jones, 1972; Rajaraman and Ullman, 2011). Since TF shows the occurrence of each term, IDF offsets the weight of common terms from TF. A reweighing and IDF-Smooth function can avoid zero divisions.

(3) Since latent semantics is based on spectral clustering, taking (a truncated) Singular Value Decomposition (SVD) of the affinity matrix ($A$) can compute the corresponding eigenpairs. The result of this matrix factorization, $L = U\Sigma V^T$, calculates that the columns of $U$ and $V^T$ are the eigenvectors for the word-by-sentence matrix. $\Sigma$ is a diagonal matrix with non-zero singular values. Therefore, finding the $K$ largest eigenvectors could eliminate the noise and approximate the solution of the optimal cut in Equation 2.2.

$$\underbrace{L}_{m\times n} = \underbrace{U}_{m\times m}\underbrace{\Sigma}_{m\times n}\underbrace{V^T}_{n\times n} \tag{2.1}$$

then it can be approximated by a truncated representation with $k$ components,

$$\hat{L} = \sum_{i=1}^{k} u_i \sigma_i v_i^T \tag{2.2}$$

where $m$ is the number of requirements, $n$ is the number of unique words, and $k$ is the predominant number of singular values.

By adjusting the optimal number of $k$ values to create a filter, we could reduce the noise from datasets. One way to determine the $k$ value is to use the Power Method by computing the eigengap of each eigenvalue until convergence (Mavroeidis, 2011; Ng et al., 2002; Zelnik-Manor and Perona, 2005). After sorting the $k$th largest eigenvalues in descending order, we used eigenvectors to determine which words are similar to each other for each concept. Then we can list and compare the $p$ number of common words/phrases to determine for each intrinsic concept. Other methods to group similar words/data can be determined by using Bayes's theorem (Fu et al., 2013) or k-means (Ng et al., 2002; Von Luxburg, 2007) to construct each group.

(4) In comparison to any two sentences from a DSM (a sentence-by-sentence matrix), evaluating eigenpairs or computing the cosine similarity kernel can extract the hidden content from requirement documents with similar terms. Analyzing the eigenvectors of the word-by-sentence matrix is a common technique to study the connections among each sentence through unique words. Likewise, cosine similarity is another approach to measure the similarity between documents, and it could compute coherence values ranging from independent to correlative with -1 to 1, respectively. The value 0 indicates that the two documents are orthogonal in vector space and that they contain no shared words.

### Spectral Clustering

To generalize topics for design documents, spectral clustering can be potentially combined with LSA. In graph theory, graph partitioning is one of the most widely used techniques for data analysis, with applications ranging from image processing to text mining (Polanco and San Juan, 2006; Wu and Leahy, 1993). The goal of a graph partition is to separate the graphs and maximize the distance of within-group connections while minimizing the number of between-group connections. Conversely, the separated groups possess the most dissimilar "patterns" (Von Luxburg, 2007). Instead of calculating the total cost of edge weights, another partition method, normalized Ncut, computes the minimum cut cost as partition criteria (Shi and Malik, 2000). Normalized spectral clustering is then used for approximating Normalized Ncuts (Mavroeidis, 2011; Von Luxburg, 2007).

Engineering requirements can be mapped into undirected graphs, while the value of affinity matrices (e.g., DSM) can represent the edges of graphs. For a given graph, each vertex represents a requirement sentence or a unique word, and the edges define the similarity between any two vertices (Wu and Leahy, 1993). The similarity could be measured by using LSA or different distance metrics (Xing et al., 2003; Zelnik-Manor and Perona, 2005). Typically, data samples exist in a high-dimensional feature space. Spectral methods can convert the high dimensional data sets to eigenspace and compute eigenvectors of the graph's Laplacian to derive clusters (Chung and Graham, 1997; Keogh and Mueen, 2010; Ng et al., 2002; Zelnik-Manor and Perona, 2005). By analyzing the eigenvectors we can discover the optimal number of groups. To choose the leading $K$ eigenvectors to separate the clusters well, the SVD is used with the $K$-means algorithm (Bach and Jordan, 2006; N. Liu and Stewart, 2010; S. Wang and Rohe, n.d.).

### System Modeling Approaches

MBE has been widely embraced by major organizations in the industry (Lubell et al., 2012). A survey has verified that MBE improves the entire system lifecycle compared to the traditional drawing-based process (Bajaj and Hedberg Jr, 2018; Rangan et al., 2005). It is important to note that MBE describes a real-time-three-dimensional digital information exchange across through product design, and it generalizes the product lifecycle into four sections: (1) product requirement modeling in complex systems, (2) mechanical design models in product data management (PDM), (3) Computer-Aided Manufacturing (CAM) models based on MTConnect data, and (4) quality inspection reports in (check) the Quality Information Framework (QIF) derived from the Quality Management System (QMS).

The current approach to identify possible problem areas and estimate the range of engineering changes within subsystems requires domain experts' judgement supported by different modeling tools (McLellan et al., 2010). The international council on systems engineering (INCOSE) defines model-based systems engineering (MBSE) as a formal modeling approach for supporting the design, analysis, validation, and verification of system requirements. In contrast to document-based information exchange, MSBE utilizes domain models as a primary method of exchanging information. Based upon Unified Modeling Language (UML), System Modeling Language (SysML) (Haskins et al., 2006) is one of the popular modeling techniques for determining logical architecture. Through the representation of requirements, behavior, structure, and parametric correlations of the system, the SysML can facilitate decision-making activities, such as requirements analysis or architectural design. For instance, with the help of a requirement diagram, designers could view, understand, and track the propagation of changes across different specifications. Implementing SysML can assist engineers in detecting errors, defects, and potential problems in industries such as automotive (Nouacer et al., 2016) and avionics systems (Gregory et al., 2020).

The electromechanical equipment of today exhibits many characteristics of complex systems. Current practice of analyzing system-of-systems problems often requires a combination of software (SysML) and domain experts to identify, solve, and verify the relations between physical components and functions (Eng et al., 2017; Mørkeberg Torry-Smith et al., 2013). However, tracing the engineering changes using traditional graphical representations of the requirement management software may not accurately represent the higher order change propagation. For modeling such change behavior, the DSM can be utilized to predict engineering changes in requirements documents. Such correlations are typically many-to-many in nature, where engineering changes propagate between high-level requirements (operational requirements) and low-level requirements (technical requirements) (Cellucci, 2008; Hull et al., 2005). Rather than graphically represent information for each requirement, DSM maps correlations using handcrafted features designed by domain experts (Browning, 2015; B. W. Morkos, 2012). Through interpretable correlations, engineers can track changes in requirements documents.

Furthermore, as requirements are constantly evolving throughout a project, maintaining a large requirement management system can be challenging (B. Morkos et al., 2019; B. Morkos et al., 2010; J. D. Summers et al., 2014). To overcome this problem, ARCPP has been developed as a tool that predicts requirements changes in real time by using the physical (nouns) and functional (verbs) patterns derived from each sentence (Hein et al., 2015; Htet Hein et al., 2017; B. Morkos et al., 2012). As compared with other approaches, ARCPP addresses the challenge of unanticipated requirements changes within requirements documents. Other than predictive models, many graphical packages and SysML models have helped engineers gain a better understanding of processes and visualize the relationship between engineers' products and stakeholder needs.

## Requirements in Smart Manufacturing

For smart manufacturing to achieve higher production, higher quality, and cost-effective rates, unstandardized or unstructured data such as requirements must be reevaluated (L. Wang et al., 2021). With the integration of data science and manufacturing, the direction of requirement management in PLM will undergo a paradigm shift. Future cloud manufacturing (CMfg) will be dependent on customers' service requirements (Tao et al., 2015), such as decentralized production 3D printing. Users will choose from multiple cloud services based on their needs, and the service will offer the most optimal options to reduce the cost.

A blockchain-based PLM system is proposed to improve data security, allowing individual designers to store decentralized design documents across multiple stakeholders (X. Liu et al., 2020). Design information, including text, images (e.g., drawings) and 3D model requirements, will be stored in cloud databases. Upon adding a new block to the network, all systems from stakeholders will automatically verify and update synchronously based on historical records. As design manufacturing requirements continuously evolve, the direction of product requirements will increase in variety, quality, and service while maximizing the satisfaction of customers.

### Requirement Datasets

There are three in-house industrial requirement datasets implemented in this paper. First, project 1 is designed for a manufacturing company to design, program, and install threading line equipment. It contains seventeen general sections varying from general descriptions to technical specifications. Second, project 2 depicts the design specifications of yarns on a spool through an automated creel system. In the textile industry, creel is designed to hold a comb of yarn. The project 3 consists of the design of an exhaust gas recirculation bypass flap with an accompanying electrical design. It is important to note that each project consists of unstructured natural language data containing different sentence lengths and vocabulary embedded with domain-specific knowledge.

## 2.2 Topic Modeling

LDA is a generative probabilistic model introduced (Blei et al., 2003) for representing discrete data, commonly in the form of a collection of documents. LDA arose as an improvement upon Hoffman's probabilistic latent semantic indexing (pLSI) model (Hofmann, 1999); whereas pLSI only provided a probabilistic model at the level of topics, LDA incorporates an additional probabilistic model at the documents level. LDA assumes that each document in a corpus has a hidden, underlying structure. Each word is generated by first randomly selecting a topic according to the requirement's topic composition and then randomly selecting a word according to the chosen topic's word composition (Blei, 2012). Every document is modeled as a mixture of $k$ latent topics, where each topic is defined by a multinomial distribution over $N$ unique words (Blei et al., 2003). This study will implement LDA in the previously unexplored domain of requirements documents.

After values for $\alpha$, $\beta$, and $k$ are assigned, topic modeling algorithms can identify the most likely topic composition for each requirement in a requirements document. Usually, either variational or sampling-based methods (Blei, 2012) are used to solve the LDA inference problem. For this study, we use collapsed Gibbs sampling (CGS), a widely used sampling-based method. CGS is a Markov-chain Monte Carlo method first applied to LDA (Griffiths and Steyvers, 2004). We use CGS to iteratively determine the most appropriate topic for each word given 1) the two Dirichlet hyperparameters, 2) the requirement's current distribution over topics, and 3) the distribution of that word over topics for the entire requirements document. CGS accomplishes this by approximating an intractable sum, known as the posterior, over a set number of iterations (Blei, 2012). While this paper utilizes and presents the mathematical algorithms derived from CGS, a rigorous mathematical description of the sampling is detailed here (Griffiths and Steyvers, 2004; Porteous et al., 2008; Xiao and Stibor, 2010).

A fictitious example presents how LDA works to depict the hidden topic structure for 3D-printer requirements in Figure 3.20. The topic and word simplexes contain Dirichlet distributions of topic compositions for each requirement and word compositions for each topic, respectively. In the topic simplex, each corner represents a topic, and each dot represents a requirement.

The multinomial distribution of each requirement over these three topics can be visualized as the proximity of each dot to the three corners. The included histogram indicates the probability values corresponding with the example requirement's proximity to each topic. The hyperparameter, $\alpha$, influences how requirements are dispersed throughout the simplex. For values of $\alpha$ smaller than one, requirements are more likely to be focused around one of the three topics, and when $\alpha$ is set equal to 1, requirements are evenly distributed throughout the simplex. In the word simplex, the hyperparameter, $\beta$, is similarly used to model vocabulary compositions for each topic. The charts below the word simplex indicate the distribution of words for each topic.

Once a hidden topic structure is identified, designers must interpret an appropriate label for each topic based on its distribution over words. Our example in Figure 3.20 includes LDA's resulting word distributions for each topic, with topics initially labeled as Topic 1, Topic 2, and Topic 3. Then, designers could interpret Topic 1 as "Printer Head," Topic 2 as "Extrusion Settings," and Topic 3 as "Build Material" based on the proportions of each word in the topic. These interpretations are subjective and should consequently be performed by domain experts. Note that the number of topics and words will not be equal in practice; typically, there are many more words than topics, resulting in greater differentiation between each topic's word composition than is seen in this example. Additionally, words from the example requirement that would typically be included in the LDA process, such as "printer," "rate," and "buildup," are ignored for simplicity.

After obtaining results, LDA's performance must be evaluated. Perplexity and coherence are applied as measures, which respectively assess the generalization of a trained probabilistic model to an unforeseen sample (Teh et

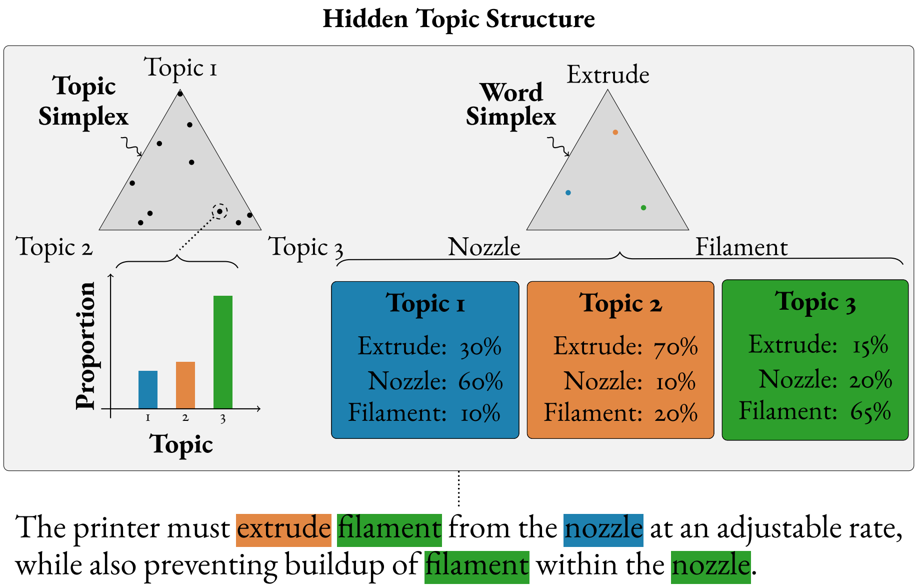


Figure 2.1: Conceptual Example Of Hidden Topic Structure

al., 2007) and the semantic similarities among words. Perplexity is commonly used to evaluate linguistic models, with a low score indicating a high degree of generalization. Intuitively, lower perplexity in LDA represents a more robust generalization performance. Contrary to perplexity, coherence tends to align well with human judgment and is often used to determine the number of topics. To ensure the effectiveness of the results, domain experts validate the generated topics.

Within many existing techniques for performing semantic analysis, LDA is widely used to extract latent topics from a collection of documents (Blei, 2012). The literature indicates that LDA overcomes some limitations of precursor thematic analysis models, such as TF-IDF, LSI, pLSI, and NMF (Blei, 2012; de Paulo Faleiros and de Andrade Lopes, 2016; Gyory et al., 2021). In LDA, each document is assumed to be a collection of topics, and each word contributes to multiple topics with varying probabilities. The LDA is a generative probabilistic model that creates topics based on large observations and predicts the topic composition of unobserved documents. The number of interpretable topics is typically predetermined by domain experts. However, LDA has limited capabilities for generalizing topics from short documents with few words, such as tweets and Reddit posts. Several variations of LDA have been developed, such as short text clustering (Qiang et al., 2020; Yin and Wang, 2014), to enhance the performance of topic modeling in smaller datasets. The Gibbs sampling algorithm for the Dirichlet multinomial mixture (GSDMM) (Yin and Wang, 2014) is a variation of LDA. The movie group process analogy provides an insightful understanding of the GSDMM, in which students can be clustered into $K$ tables based upon their common interests in movies. As each movie title can appear only once, the clustered results strike a balance between completeness and homogeneity. As a result of different model assumptions, both models can generalize words into topics with varying levels of performance.

## 2.3 Relevance to Design Research and Practice

Improvements or automation to requirements management could change how requirements are elicited, documented, and verified. Currently, requirements management includes the documentation of requirements with minimal tracing and exists mostly within its own domain of requirements (e.g., it does not build relationships with design tasks or activities outside of requirements). Further, understanding requirements from a topical perspective may provide designers and managers with a mechanism for ensuring requirements completeness. Topics may serve to appropriate requirements into pertinent design groups. For instance, a suspension team could receive requirements related to components of shock absorbers.

In design research, LDA is studied in design group cognition (Gyory et al., 2020), idea generation (Ahmed and Fuge, 2018), and product attributes (Joung and Kim, 2021). The concept of topic generation can also present a mechanism for requirements automation that can be performed early in the design process and does not require specific designer expertise once when managing the requirements documents. In doing so, topics (and their associated requirements) could determine resource allocations and inform design-space exploration. Designers can further interpret latent stakeholder needs and interests based on the generated topics to determine engineering requirements and improve project success.

## 2.4 Overview of BERT Architecture

Integrating a pre-trained embedding model enhances the quality of the design topics and improves the model's overall performance. BERT is a transformer-based bidirectional model used for natural language classification, question answering, language inference, and sentence similarity tasks (Devlin et al., 2018). BERT models are typically trained by using more than 3.3 billion words retrieved from open online libraries, such as Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and Wikipedia articles (Dor et al., 2018). For a sentence similarity task, BERT model would require $n(n-1)/2$ inference computations. By adding a pooling layer at the end, sentence-BERT overcomes this limitation, resulting in computations that are equal to the number of sentences analyzed. To improve performance, sentence-BERT models are pre-trained on Wikipedia and NLI datasets and fine-tuned on STSB datasets (Reimers and Gurevych, 2019).

Numerous studies have evaluated the knowledge that pre-trained BERT models can extract from large linguistic datasets. BERT can encode the general knowledge in both syntactic and semantic representations (Lin et al., 2019). The concept of syntactic knowledge relates to the relationships among words to form a meaningful sentence. The BERT model encodes linguistic information as hierarchical structures as opposed to linear structures to create syntactic dependencies. However, this knowledge dependency has not yet been fully understood. On a higher level of abstraction, the self-attention mechanism represents partial syntactic structure via attention weights. Using these attention heads, a syntactic tree can be built, often packed inside a [CLS] token, to solve a prediction task. The notion of semantic knowledge concerns the meaning of words and sentences. Similar to convolutional neural networks, the lower layers contain low-level features, while the higher layers represent semantic features (Jawahar et al., 2019). Incorporating a pre-trained Sentence-BERT model provides general language representation to facilitate syntactic and semantic comprehension of requirement documents. As pre-train BERT-based models can handle a range of tasks, relying solely on the sentence-BERT model may not be sufficient to capture design topics from domain-specific requirements documents. The idea of combining LDA and Sentence-BERT can mitigate the disadvantages of using a single model.

## 2.5 Techniques for Supporting Joint Embedding

Besides textual documents, other types of data can also be linked and jointly represented with requirements. There are times when different types of data might not be properly collected or saved during the product design process. To compensate for this loss, a synthetic image dataset can mimic real-world data.

## Image Scraping

A digital thread is more than just digital transformation - it is the ability to extract useful information from different types of data sources. The use of image retrieval techniques could potentially contribute to building such digital threads and to correlating images and text, which are widely used in social web applications. As image data may not always be available, image retrieval is used to search and collect online images. Image retrieval can be divided into three categories: text-based image retrieval (TBIR), content-based image retrieval (CBIR), and semantic-based image retrieval (SBIR). For example, search engines like Google rely on TBIR systems (van Gemert, 2003). Through a query, text-based retrieval can be simplified into a keyword-based search, and the returned results can be visualized as images with semantic similarity (Datta et al., 2008).

Combined with TBIR systems, web scraping is a technique which can collect information from Google. Such scraping tasks include reading HTML links, image files, and audio records (Mitchell, 2018). The challenge of collecting information online involves complicated website structures and bot access as known as the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). Many libraries are built to aid designers to automatically download images based on queries, such as the Selenium, the Google-image-download, and the Beautiful Soup libraries. These tools allow users to search and modify the raw content through appropriate parsers using Python. Based on targeted image URL links, information is downloaded for further analysis.

## 2.6 Point Cloud Classification

Among the major challenges of classifying 3D models is the improvement of algorithm efficiency and real-time execution. Several deep neural networks have been developed to address this issue, including PointNet (Qi, Su, et al., 2017), DGCNN (Y. Wang et al., 2019), and SimpleView (Goyal et al., 2021). As a result of its unified architecture, PointNet utilizes the permutation invariance of points and processes each independently with a symmetric function that aggregates the features. EdgeConv is a block introduced by DGCNN that exploits both local and global shape properties for each point as topological information. In SimpleView, 3D point clouds are converted into 2D depth images by utilizing a projection-based method. Generally, deep neural models require a lengthy training period and are dependent on a number of parameters to achieve higher accuracy. To date, these techniques do not convert feature space into semantic embedding space for the classification of 3D models.

Current literature of automatically perform object classification or part segmentation has mainly focus on deep learning approaches (Bello et al., 2020; Y. Liu, Fan, Meng, et al., 2019; Y. Liu, Fan, Xiang, et al., 2019; Qi, Su, et al., 2017; Qi, Yi, et al., 2017; Y. Wang et al., 2019). The disadvantages of deep learning approaches are expensive to train and the hidden weights are not interpretable. Many shallow learning techniques using feature extraction can be implemented in point cloud applications to develop more computationally efficient models and overcome certain model limitations. Semantic autoencoding (SAE) is a technique for learning project functions from feature spaces to latent spaces (Kodirov et al., 2017). SAE is a zero-shot learning technique that scales large-scale datasets by computing the Sylvester equation [10](Bartels and Stewart, 1972). In point-cloud representations, this approach has been widely implemented and is a solution to the problem of recognizing 3D objects.

Recent advances in computing technology have made it possible for manufacturers to combine point clouds and 3D computer vision techniques to systemize domain experts' knowledge for building automation systems. Current electro-mechanical systems are capable of recognizing 3D characteristics of objects ranging in size from nanometers to kilometers. For smart manufacturing applications, each data point in a point cloud contains spatial information that provides precise position information. While there are many innovative algorithms for processing 2D images, 3D object recognition has not yet been fully explored. With high precision models, designers can quickly identify objectives or mechanical sub-assemblies, detect manufacturing defects (Lyu et al., 2021; Nouacer et al., 2016), and reconstruct CAD models for reverse engineering purposes (Vafaeesefat and ElMaraghy, 1999). This study is primarily focused on object recognition to build such a framework. Further, the work could lead to an automatic system to detect mechanical subassemblies.

## Mechanical Geometric Modeling

Geometric modeling has a significant impact on the design and manufacturing of products. The paradigm shift from engineering drawings to computer-aided design models has significantly changed the way engineering design products are manufactured and analyzed. The use of CAD and

[10] As the Sylvester equation is based on the size of the feature space rather than the number of samples, this method can significantly improve the computational efficiency of point cloud applications.

computer-aided engineering (CAE) has become common practice in the fields of design, manufacturing, and quality inspection. By visualizing virtual products during the early design phases of a product, CAD models can assist in ensuring high quality and accuracy. Many 3D representations, such as point clouds, 3D meshes, and voxels, are developed to study existing structures as 3D CAD models become more prevalent in engineering applications.

## Computer-aided Modeling

Since the 3D CAD model replaced engineering sketches/drawings, digital documents have improved the reusability, accessibility, and quality of engineering model designs (Frechette, 2011; Karima et al., 1985; Veisz et al., 2012). 3D CAD representation contains a set of distinct parts, such as geometric objects generated as CAD format, including completed product components and assemblies (e.g., product materials and manufacturing information) (T. Hedberg et al., 2016; Wardhani and Xu, 2016). In industrial practice, design engineers interpret system requirements and create CAD models for every step of the product lifecycles. Any product design modification would result in a time-consuming procedure to mitigate potential system failure (B. Morkos et al., 2012). In response, our goal is to associate CAD models with corresponding requirements and reduce the liability of changes in a complex system. To numerically represent CAD models, different designs can be represented using a number of geometric techniques.

- **Voxel** are often viewed as 3D pixels for volumetric data, where 3D ShapeNet (Wu et al., 2015) extract information from 2.5D depth images to recreate the 3D shapes using cuboids. The advantage of voxel provides the flexibility to generate high accuracy 3D building block for object recognition.

- **3D mesh** is another method to visual objects in terms of polygons with vertices, edges, and faces. High-fidelity models require many polygons, which increases memory usage. It is common for mesh representations to be sensitive to irregular elements, making them difficult to edit and analyze.

- **Point clouds** contains a finite number of dots (e.g., including the values of each point for X, Y, Z coordinates) to represent a 3D object. Typically, point clouds data are collected using Lidar scanner or can be converted from other types of data, such as OFF or STEP files. In practice, point clouds data are often implemented for its high accuracy and low memory usage.

Point clouds are computed efficiently by converting the data into a common standard format such as HDF5. In most point cloud benchmark datasets, such as ShapeNetCore (Yi et al., 2016) and ModelNet40 (Wu et al., 2015), 1024 random points are sampled for each model and normalized into a unit sphere. An individual point contains only (x, y, z) coordinates, and a label identifies the component groups. A comparison between two different datasets is shown in Table 2.2 below.

| | # of Classes | # of Samples | # of Parts |
|---|---|---|---|
| ShapeNetCore | 16 | 16,881 | 2 - 6 |
| ModelNet40 | 40 | 12,311 | - |

Table 2.2: A Comparison Of Two Popular 3D Datasets

**Clustering in Computer-aided Model**. The goal of clustering CAD models is to match the subcomponents from requirements. Research in this area has paid little attention to building an appropriate number of clusters (e.g., sub-assemblies) corresponding to requirements. In the current practice, online outsource cloud platforms enable people to manually label CAD components and to have these labels verified by domain experts for the purpose of building machine learning datasets. Since the mechanical design space is vast, recognizing different types of CAD subassemblies can be challenging and requires a large amount of data to train. Aside from the challenges associated with building such datasets, there is also the issue of matching the CAD subassemblies with the corresponding requirement concepts (e.g., topics). Particularly, we realized that since requirements might not explicitly describe how designers should create each small component for CAD models, and a mismatched groups could occur. As a result, building a model-based approach remains the most practical solution.

## Meta-Learning

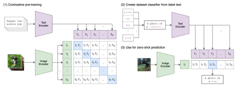
Meta-learning (Finn et al., 2017) consists of learning multiple tasks simultaneously to train a model without adding any additional parameters. In comparison to a standard stochastic gradient descent (SGD) method, meta-learning updates gradient parameters based on the number of tasks. As a result, this step requires an additional backward pass to compute Hessian-vector products using various Python packages. As an extension to the original study of $K$-shot learning settings, zero-shot meta-learning has gained popularity in recent years (Mohammadi et al., 2019).

## Joint Embedding

As digital threads become more prevalent in industry, computer vision techniques are making their way into other fields, such as manufacturing. Joint embedding learning involves mapping different types of information, such as images, texts, speech, and video, into a common latent vector space. The most common research challenges occur in the areas such as bi-directional image and text retrieval (Faghri et al., 2017; L. Wang et al., 2018), visual question answering (Antol et al., 2015), and image captioning Karpathy and Fei-Fei, 2015. Previously, Canonical Correlation Analysis (Gong et al., 2014; Hardoon et al., 2004) is used to find the linear combination of image and textual data that maximizes the correlation between image-text pairs. Using this method, correlations can be built between images (e.g., engineering drawings or photos) and text documents (e.g., requirements documents, interview dialogues, or project descriptions). However, this approach can present correlated image and text features at a high memory cost. A variety of loss functions were developed to overcome this problem, including margin-based loss (Frome et al., 2013), bi-directional ranking loss (L. Wang et al., 2018), triplet loss function (Schroff et al., 2015), and multi-class $N$-pair loss (Sohn, 2016).

A joint embedding model, CLIP [II], which is trained on 400 million images and texts from publicly available datasets, uses supervised zero-shot learning. The zero-shot learning approach is characterized by the fact that no classes are presented during testing that were presented during training (Socher et al., 2013). As the CLIP model can be boiled down to image and text embeddings during training, this structure allows diverse types of neural networks to be applied to the image and text encoders. To scale down the number of parameters, the image encoder typically employs either the vision transformer

[II] CLIP has a flexible architecture,



(Radford et al., 2021) where different encoders can be used to train various types of data jointly. A pre-trained CLIP learns across a wide range of tasks and is often used to fine-tune a task domain specific task.

(ViT) (Dosovitskiy et al., 2020) or ResNet (He et al., 2016). Several common building blocks are used in the construction of a text encoder, including the BERT or transformer-base models (Sanh et al., 2019). The BERT model supports a maximum sequence length of 512 tokens, while the transformer model truncates the sequence length to 76 tokens to improve computation efficiency. Depending on the type of application, a simpler model might achieve a better generalization performance. Meanwhile, a cosine similarity is then calculated among images and texts and evaluated with a chosen loss function, such as cross-entropy loss. As a part of testing, the zero-shot CLIP model provided more reliable results for out-of-distribution image prediction.

# Chapter 3

# Research Methodology and Findings

Based on the current literature review, this chapter discusses three proposed methodologies for contributing to digital threads.

- Requirement Topics: A topic modeling approach is implemented on requirements documents and creates a layer of topics to assist designers in determining engineering changes.
- Point Cloud Classification: A neural network model is proposed to recognize point cloud representations in mechanical design models.
- Joint Representation: Through the training of a foundation model, this approach extends the previous two domains knowledge by learning the associations between requirements and CAD. By fine-tuning CLIPs, this approach can leverage both domain-specific and general knowledge.

The goal of this chapter is to present details of the implementation of requirements topics, point cloud classification, and joint representation of diverse types of information. Each section begins with a description of the proposed methodology, followed by a discussion of the results.

## 3.1 Experimental Study

Before the implementation of LDA, a proof-of-concept example is carried out to show that topics can be extracted from requirements documents using LSA. For project 1, we only considered technical requirements that are related to the mechanical components and operations, such as operations, general equipment concepts, benefits, description of equipment supplied, and welded tube scope of supply with a total of 247 sentences. The rest of requirements are considered as non-technical (description of project, project specifications, shipping, installation and start up, documentation, training, project management, design planning and design control, acceptance, preliminary project schedule, delivery, notes and exceptions, warranty containing a total of 104 sentences) requirements.

Based on the technical requirements, the word-by-sentence matrix has been converted into eigenspace, and each concept contains unique words ($p$ = 8) that have a tight intrinsic relationship, as displayed in Table 3.1. Upon our initial observation, certain unique words repeat several times within or across concepts. For instance, the word "pipe" is often used as "pipe stops" or refers to a physical pipe. Also, most of the unique words captured within those five concepts are nouns, and previous research has indicated that nouns are more likely to instigate propagation in requirements than verbs (Hein et al., 2015). For each concept, we hypothesize that a frequency analysis could determine the most correlated requirements. We tested this idea by comparing each concept with requirements. For instance, we assumed the designers need to modify the "station" design in concept 1. In this case, "station" refers to a threading station in Req. 5.1.10.1. After the keyword search, we randomly picked four requirement sentences, including "station," shown in Table 3.2. The results of each sentence contain a different number of unique words highlighted in bold. Since Req No: 5.1.4.2.3.3 and 4.2.8.2 have less than three unique words from concept 1, we can design a minimum threshold ($\theta \geq 4$ words) to determine the affected requirements. For Req No: 5.1.4.2.3.3 and 4.2.8.2, those two requirements have been influenced by the initial requirement change from the cognitive perspective. This finding has been verified with ground truth. Furthermore, since Req No: 3.49 contains key words from concept 2 as well, there will be many overlaps when cluster requirements into groups. If each concept can represent one dimension, the total number of concepts will depend on the parameter, $K$. Therefore, mapping the entire requirement document could be computationally expensive and hard to visualize.

Table 3.1: The Five Concepts Generated From LSA With Its Unique Words For Project 1

| **Concept 1:** | **Concept 2:** | **Concept 3:** | **Concept 4:** | **Concept 5:** |
|---|---|---|---|---|
| end | lift | blast | next | member |
| pipe | stop | end blast | vroller | structural member |
| station | pipe | end end | return | weld structural |
| box | pipe stop | blast station | allow pipe | structural |
| box end | fix | end | next station | station frame |
| threading | fix pipe | radial | allow | return |
| lift | paddle | radial roller | pipe | frame |
| stop | paddle lift | transfer | transfer | construct weld |

In the same manner as with project 1, project 2 has been subjected to the same threshold value. The results of the calculation are displayed in Table 3.3. For demonstration purposes, a unique word "stainless" has been chosen

Table 3.2: The Selected Requirements Highlighted With Unique Words From Concept 1

| Req No: | Descriptions: |
|---|---|
| 5.1.4.2.3.3 | After **pipe** is in position within blasting **station**, radial rollers rise and V-rollers lower. |
| 4.2.8.2 | HMI provides overall view of status of line and **station** by **station** statuses. |
| 3.49 | **Pipe** is **lifted** off of adjustable **pipe stops** at **thread** inspection table and lowered onto **station** V-rollers by paddle **lifts**. |
| 5.1.10.1 | **Station** design is identical to **box end threading station**. |

from concept 2 shown in Table 3.4. Based on the content of this project, "stainless", "stainless steel", and "steel" have similar meanings and can be observed in all four selected requirements. After a frequency analysis, the first two requirements are most related. Thus, with any modification on stainless material, both first and second requirements are more likely to be affected. Note that not all the unique words are useful, and there is some noise due to incorrect preprocessing, such as "end end" in concept 3. For this reason, a verification step is necessary to ensure the quality of each concept.

Table 3.3: The Five Concepts Generated From LSA With Its Unique Words For Project 2

| Concept 1: | Concept 2: | Concept 3: | Concept 4: | Concept 5: |
|---|---|---|---|---|
| datum | equipment | list | design maximum | supplier |
| limit follow | stainless | requirement | equipment system | approval |
| document datum | stainless steel | engineering | including | following |
| document | steel | datum list | including limit | approve sub |
| limit | display | datum requirement | maximum personnel | following purchaser |
| follow | display follow | engineering document | personnel safety | prior start |
| drawing | equipment item | form | safety including | purchaser approval |
| purchaser document | fit securely | form limit | supplier equipment | start fabrication |

A proof-of-concept study conducted on two industrial projects has demonstrated that terms with intrinsic correlations can be grouped together using LSA. Following this study, we implemented LDA to study the latent topics from the requirement documents.

Table 3.4: The Selected Requirements Highlighted With Unique Words From Concept 2

| Descriptions: |
|---|
| Major **equipment** items supplied by the Supplier shall be fitted with a securely mounted **stainless steel** nameplate **displaying** the following information: Manufacturer's model and type number. |
| Fabricated **stainless steel** shall be L-grade **stainless steel** unless otherwise noted. |
| Yarn guides shall have a one inch outside diameter ceramic eyelet on a **stainless steel** plate Specifications for eyelets will be provided by Purchaser. |
| **Stainless steel** and plated surfaces shall not be painted, unless otherwise specified. |

## 3.2 Implementation of LDA

As LDA is efficient in uncovering latent topics, a case study will present each requirement project's topics distribution. This section describes the method of the case study in three phases: (1) requirement text preprocessing, (2) LDA collapsed Gibbs sampling, and (3) hyperparameter tuning for LDA.

### Phase I: Data Preprocessing

As requirements documents vary in style and format depending on the industry and company, the performance of topics modeling may differ for each corpus. The requirements documents for three different industrial design projects (named Project 1, Project 2, and Project 3) (B. Morkos et al., 2012) were selected for analysis in this study. Project 1 involves designing, manufacturing, programming, and installing threading line equipment and contains 350 requirements. Within the threading equipment, most stations share standard mechanical components, which causes repetitive words. Second, Project 2 specifies the design of an automated creel system, which is a piece of equipment used in the textile industry to secure yarn combs while weaving fabrics. The Project 2 requirements document contains 160 sentences. Third, Project 3 describes electrical cabinets and enclosures with operator panel interface equipment and includes 247 requirements. Each project contains different non-alphanumeric characters and various ratios of FRs and NFRs. Based on the number of NFRs, projects can be sorted into 2, 1, and 3 in descending order.

A data preprocessing procedure is designed to reduce the noise for three unstructured requirement datasets. This step is frequent practice in NLP to improve algorithmic performance by filtering out the insignificant words (Joung and Kim, 2021). Both stopwords (e.g., "shall," "etc.," or "must") and non-alphanumeric characters (e.g., "-" or "%") are eliminated by using NLTK's package[12]. After randomly shuffling each document 10 times, the remaining English words are lowercased, tokenized, and lemmatized to create a vocabulary of unique words. As requirements are formal writing sentences, each word assumes a single base form. A summary of each requirement corpus is presented in Table 3.5.

Table 3.5: Requirement Corpus Statistics

| Project | Number of Requirements | Avg. Tokens per Requirement | Tokens | Vocab |
|---|---|---|---|---|
| 1 | 350 | 119.38 ± 7.59 | 41782 | 793 |
| 2 | 160 | 204.00 ± 6.00 | 32641 | 806 |
| 3 | 247 | 144.31 ± 13.47 | 35645 | 1051 |

## Phase II: LDA Collapsed Gibbs Sampling

An LDA model (Blei et al., 2003) is applied for generating the requirement topics after preprocessing the text data. In this study, each requirement dataset is a corpus, and every FR or NFR is treated as an individual unlabeled document. To solve LDA inference for a corpus in equation 3.2, the collapsed Gibbs sampling method is applied to estimate the values of latent variables. A word-topic matrix ($\phi_{nk}$) is initialized after assigning a random topic to each

[12] https://www.nltk.org

word based on multinomial distribution. For each document $d$, we draw a random proportion from the document-topic matrix ($\theta_{dk}$) with the Dirichlet parameter $\alpha$. As every word $w_{dn}$ from the document-word matrix has a preassigned topic $z_{dn}$, each iteration will compute and update the word-topic matrix $\phi_{nk}$, topic probability array, and document-topic matrix $\theta_{dk}$ (Teh et al., 2007), where $\phi_{nk}$ contains the Dirichlet prior $\beta$. A maximum of 200 iterations is used to ensure model convergence. Then a perplexity measurement is calculated to represent the performance of the model generalization as $Perplexity(\mathcal{D}_{test}) = \exp\left\{\frac{\sum_{d=1}^{M} logp(\mathbf{w}_d)}{-\sum_{d=1}^{M} N_d}\right\}$, where $M$ is the total number of requirement documents.

$$p(\mathcal{D}|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left( \prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d \quad (3.1)$$

As perplexity ratings do not always correspond to human intuition (Chang et al., 2009), we calculated the coherence score, $C_v$, a measure of topicality that can be interpreted by humans, by using the *Genism* package[13], as shown in Figure 3.5. The coherence score is calculated as the average of cosine similarity, normalized pointwise mutual information, and Boolean sliding window measures for the various LDA models. A higher coherence score is more likely to correlate with human judgment and to produce meaningful topics.

## Phase III: Hyperparameters Tuning

This section adopts two different methods to estimate the appropriate number of topics. First, a line fitting technique known as the L method (Salvador and Chan, 2004) is explored for determining the optimal number of topics. This procedure is defined by closely fitting two lines to the data, and the

[13] https://pypi.org/project/gensim/

Table 3.6: After Tuning Both $\alpha$ And $\beta$ As Control Variables, The Bolded Number Indicates The Lowest Perplexity Value (Averaged Over Three Runs) Fixed At Two Hundred Iterations For Project 1.

| | $k$ = 15 | | |
|---|---|---|---|
| | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ |
| $\alpha = 0.01$ | **52.30** | 65.10 | 80.94 |
| $\alpha = 0.1$ | 71.65 | 97.80 | 115.40 |
| $\alpha = 0.2$ | 106.78 | 117.43 | 139.14 |
| $\alpha = 0.3$ | 105.53 | 135.67 | 153.37 |
| $\alpha = 0.4$ | 117.36 | 150.61 | 172.42 |
| $\alpha = 0.5$ | 127.49 | 139.20 | 189.60 |

intercept point indicates the estimated number of groups. This technique can provide a quick estimation of the number of topics without adding additional analysis. Second, one popular technique for finding the number of topics is to vary the alpha values (e.g., $\alpha = 10/k$, $\beta = 0.1$) (Griffiths and Steyvers, 2004; Jacobi et al., 2016). By varying the topic range from $k$ = 10 to 100 with an increment of 10, the lowest perplexity value indicates the appropriate number of topics on a hold-out set. Both methods are applied for each dataset to estimate the number of topics, as discussed in Section 14. After fixing the number of topics, a fine-tuned procedure has determined the best value for both $\alpha = \{\alpha | 0 < \alpha\}$ and $\beta = \{\beta | 0 < \beta < 1\}$, as shown in Table 3.6. A method of relevance measure is then applied to improve the quality of generated topics by balancing the ratio between the word-topic probability, $p(n|k)$, and the *lift*, $\frac{p(n|k)}{p(n)}$, which is a conditional distribution over marginal distribution. The results of topics' quality and interpretability are shown in the following Section 14 by adjusting the experimental value $\lambda$.

## RESULTS AND DISCUSSION

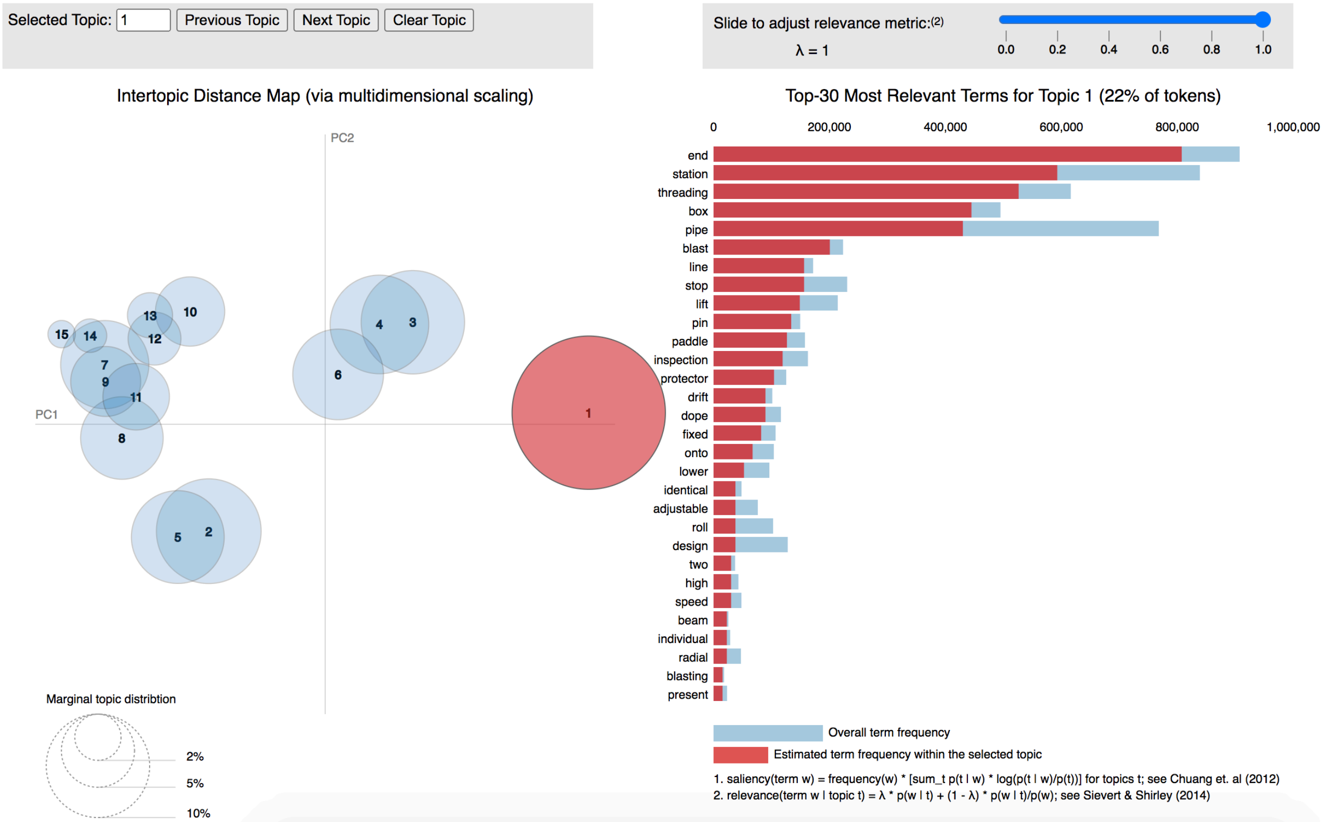
After applying the same data pipeline to all three heterogeneous requirements documents, this section discusses the findings of topic visualization, the number of topics, and the quality of sampled topics. Project 1 is presented as an example of a representative project with topic distributions.

### Topic Visualization

To address the first research question of how to represent requirements documents into topic structures, a graphical representation tool named LDAvis (Sievert and Shirley, 2014) provides an overall view of both topic and word distributions for topic interpretation. Each circle in Figure 3.1 represents a latent topic in a 2D subspace, while the topics that overlap share common words. For each topic, the top 30 most relevant words are selected based on their probability: the gray bar represents the overall term frequency in the corpus, and the red bar indicates the high-frequency terms for that topic.

As requirements often use modal verbs (e.g., must), high-frequency words are expected to appear frequently in topics and contribute little information for most topics. For this reason, designers should also consider low-frequency words for topic interpretation. In response, LDAvis provides a weighting parameter, $\lambda$, that balances this issue. Depending on the interpretation and domain knowledge of each topic, the optimal $\lambda$ value and the actual number of words to consider may vary.

The top 30 words from LDAvis can also be visualized as a WordCloud based on their word probability distribution, as shown in Figure 3.2. Font size indicates the probability of $p(n|k)$, with the most relevant words having the largest font. For instance, topic 9 can be viewed as a group of pipe dimensions



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 3.1: Fifteen Topic Word Relations For Project 1

and units. It is crucial to use such topics when redesigning products for other unit systems because requirements engineers can use such topics to identify all units and ensure that the conversions are correct.

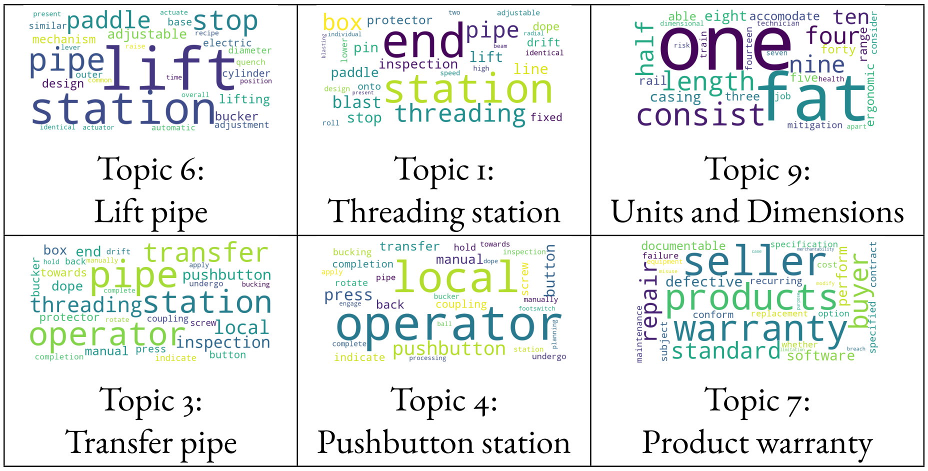


Figure 3.2: Samples Of Generated Topics With Assigned Labels For Project 1

## Quantity of Topics

The second research question studies the appropriate topics that designers should consider for each design. After performing topic visualization analysis, designers must make a trade-off decision for the number of topics, and then manually assign labels to each topic in LDA. In general, select a higher number of topics is preferable (Wallach et al., 2009). Choosing more topics is difficult to interpret, and fewer topics cannot capture all the necessary design details. Ideally, well-separated topics contain distinct word distributions orthogonal to each other in a subspace (Arun et al., 2010), meaning that topics should be diversified, and words should be distinct. We consider that the appropriate number of topics depends on either the design intent or the interpretation of designers.

Perplexity and coherence measurements are utilized to determine the optimal number of topics. As increasing the number of topics decreases the perplexity monotonically (Blei et al., 2003), we first adopt the perplexity scores for different values of $k$ in Figure 3.3. After a fine turning step, each methods (e.g., L method and $\alpha = 10/k$) was estimated at 10 and 20 topics, respectively, as shown in Figures 3.3 and 3.4. In comparison to perplexity, the coherence score indicates an optimal range of topics between 8 and 16 in Figure 3.5.

The optimal number of topics was determined by manually examining the combined range from 8 to 20. We observed that the majority of the NFRs closely overlapped each other, resulting in nearly half of the total topics. Choosing more topics would further break down topics and reduce marginal topic distributions, producing a more refined collection of topics. For example, topics 1 and 4 in Figure 3.2 refer to threading and pushbutton stations, respectively, while topics 3 and 6 represent lift and transfer motions

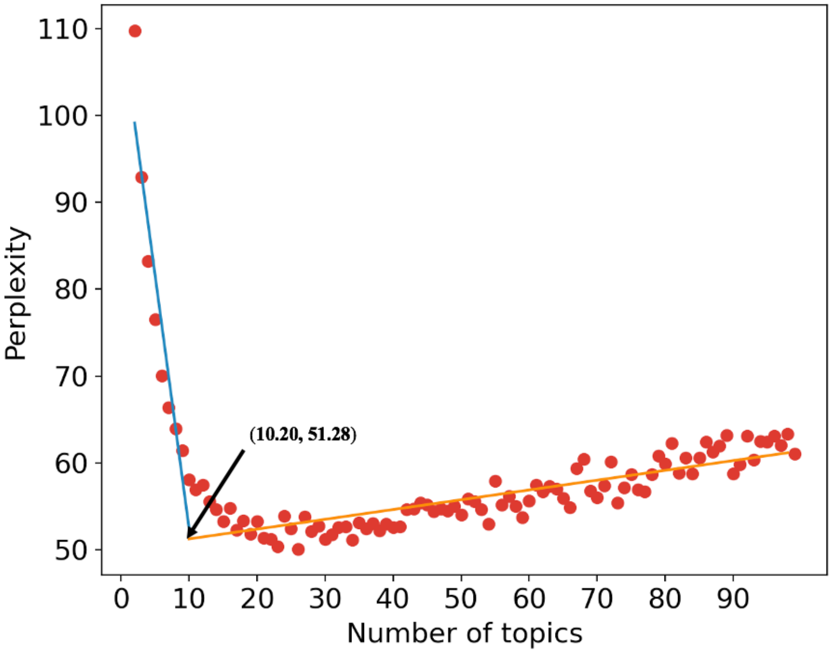


Figure 3.3: Perplexity With The Optimal Number Of Topics On Hold Out Set (Two Lines Are Fitted Using The L Method For Project 1.)

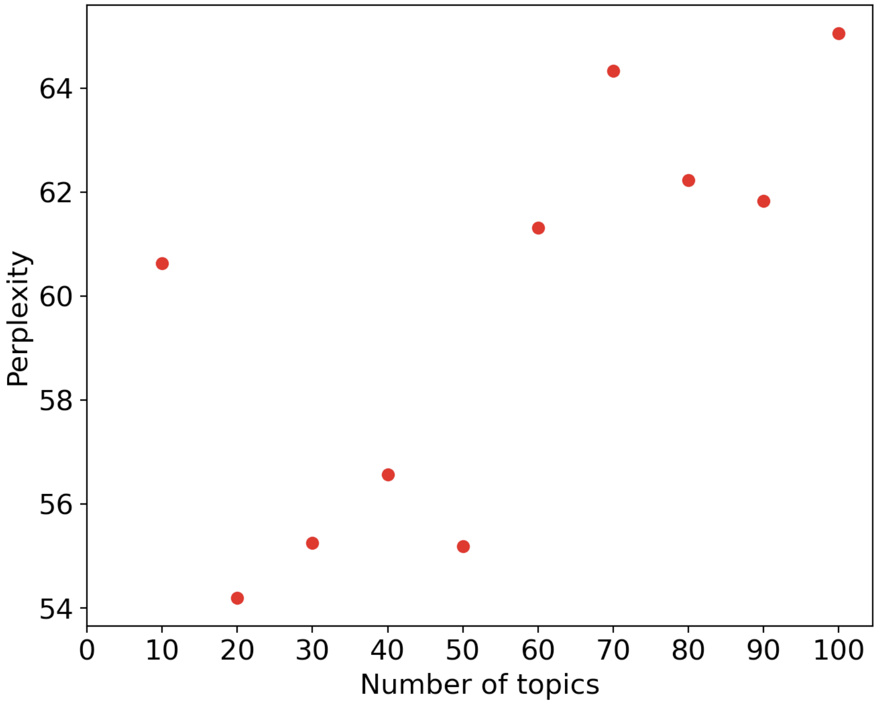


Figure 3.4: Optimal Number Of Topics Is Twenty With $\alpha = 10/k$ Method For Project 1

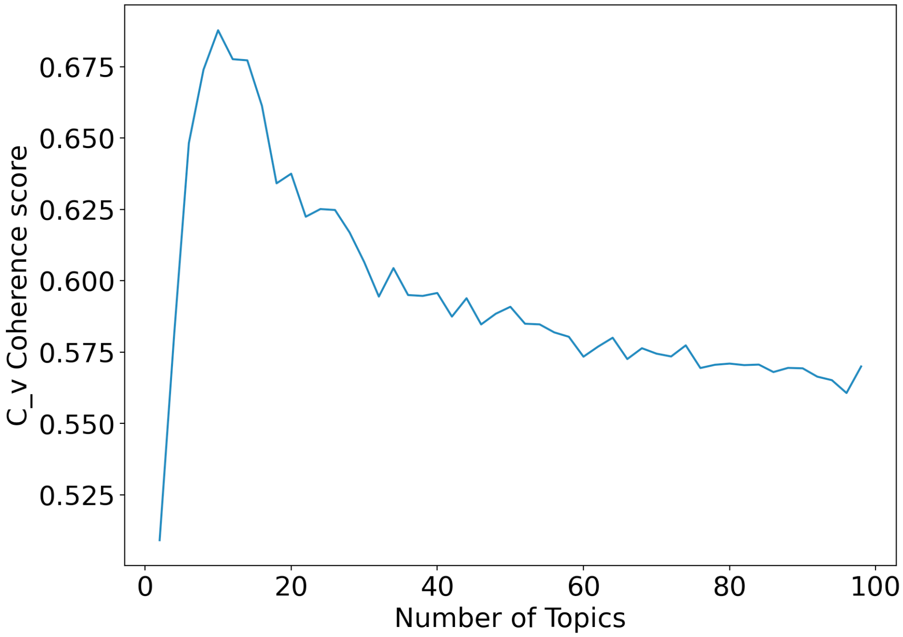


Figure 3.5: The Variation Of $C_v$ Coherence Score With The Number Of Topics.

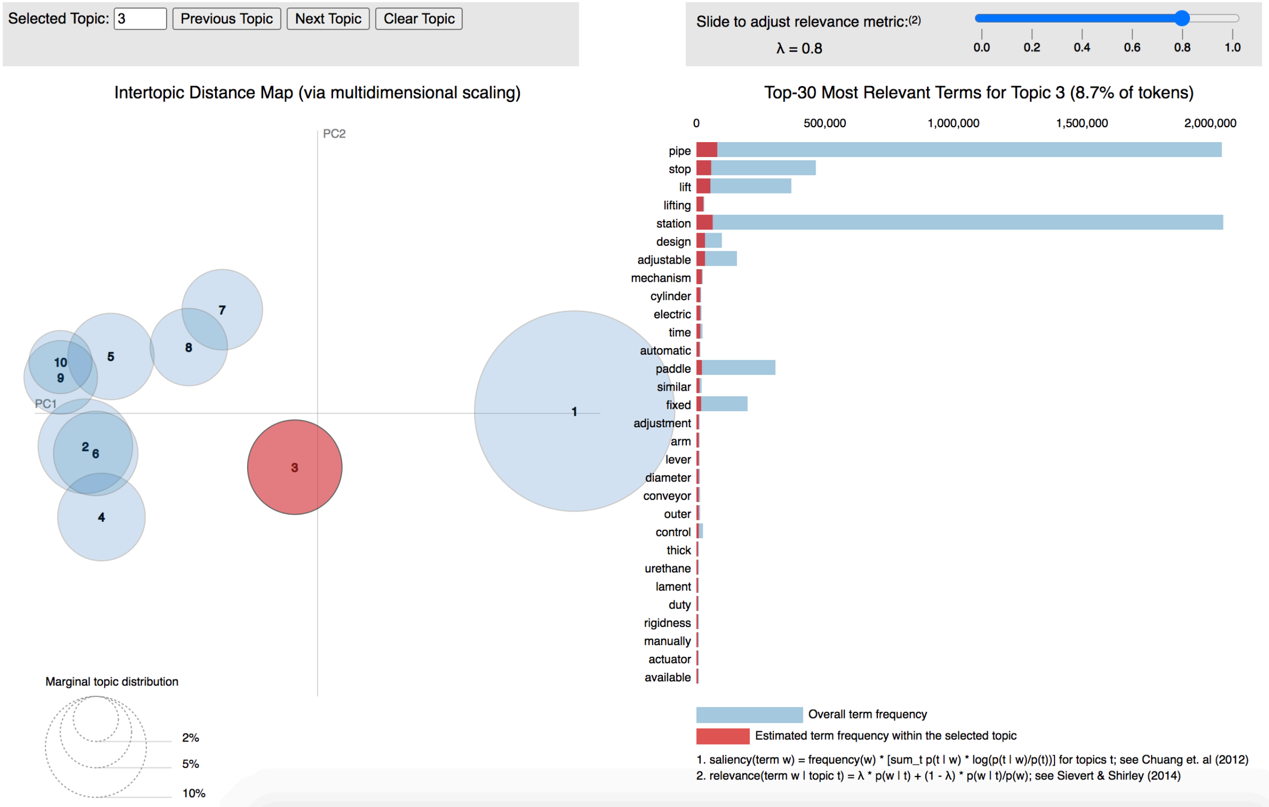
of the pipe. However, in comparing the distribution of words across various topics between 10 and 15, some of the terms used in topics 3 and 6 had a similar meaning to topic 3 in Figure 3.6, which may be regarded as a merging of topics. This is because the number of topics affects the distribution of the word-topic matrix, $\phi_{nk}$, document-topic matrix, $\theta_{dk}$, and the model parameters for the LDA when fitting the data.

**Validation** Detailed topics may not always align with human intuitions or provide valuable insights. Project 1, for instance, can be generalized into 9 different manufacturing workstations based on their functionalities by domain experts. Ideally, each generated topic should correspond to a specific station, but topic 15 performed relatively well when identifying three stations from the predetermined range. That is because the common word “station” has a relatively high co-occurrence count, such as “lifting station” or “threading station.” Selecting a higher number of topics can further segment the topics into more stations. Several of the NFRs are composed of low-frequency words and selecting fewer topics can lead to greater generalization of interpretable results. Consequently, a tradeoff decision should be based on purposeful interpretation for various corpora.

To validate the effectiveness of the generated topics, we compare the results with the predefined topics (e.g., section titles created by the industrial designers). Upon comparison, it appears that there are many overlaps between the topics, which can be applied to extract information from complex designs. As an example, one of the captured topics is “product warranty”. It contains many details regarding the duty of manufacturers and standard maintenance of products. In one of the subsections, “threading station”, various characteristics (e.g., verbs and nouns describing pipes) have been accurately captured. Other topics, such as “life pipe” shown in Table 3.6, describe the same components despite a slight difference in the titles. As a concept, such topics can accurately capture the categories as compared to the design intents of industrial partners.

Furthermore, both Project 2 and Project 3 utilize the same approach to demonstrate the robustness of the process when applied to different sets of requirements written by different authors for different systems of varying scales. Project 2 contains the highest ratio of NFRs, and project 3 consists of mostly technical requirements; each project resulted in 12 and 11 topics, respectively. The L method tends to estimate fewer topics, whereas the 10/$\alpha$ method may overestimate the number of topics and may require a fine-tuning to obtain a stable result. The generated topics of Project 2 are also capable of capturing some aspects of design information based on low-frequency words, as shown in Figure 2. Similarly, Project 3 provides further details on several

Figure 3.6: Ten Topic Word Relations For Project 1

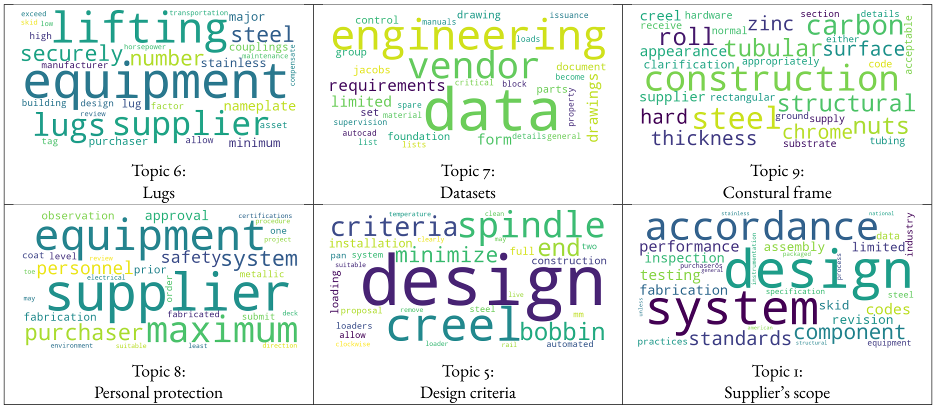

Selected Topic: 3

Slide to adjust relevance metric:(2)

λ = 0.8

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

mechanical components based on FRs in Figure 3.8. In sum, the various ratio of NFRs or FRs will affect the general outcomes of the topics. As the structure of the corpus changes, design practitioners may adjust the ratio of NFRs and FRs during data preprocessing to achieve the desired result.


Topic 6: Lugs

Topic 7: Datasets

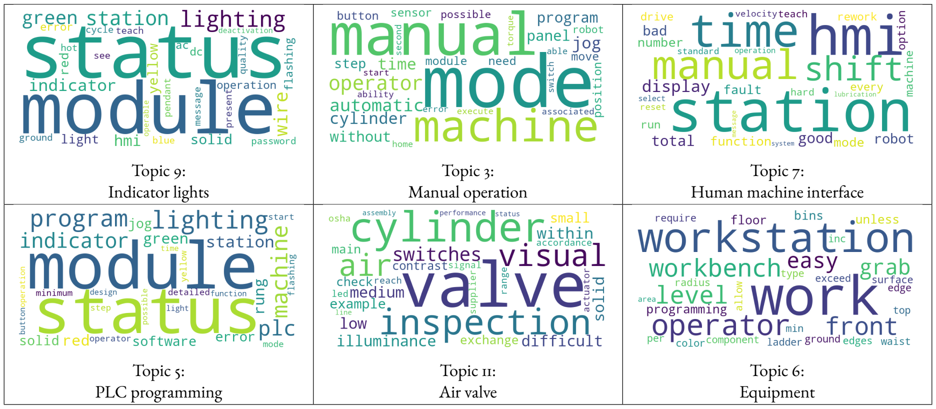
Topic 9: Construal frame

Topic 8: Personal protection

Topic 5: Design criteria

Topic 1: Supplier's scope

Figure 3.7: Samples Of Generated Topics With Assigned Labels For Project 2



Topic 9:
Indicator lights

Topic 3:
Manual operation

Topic 7:
Human machine interface

Topic 5:
PLC programming

Topic 11:
Air valve

Topic 6:
Equipment

Figure 3.8: Samples Of Generated Topics With Assigned Labels For Project 3

## Quality of Topics

For RQ3, we assessed the quality of generated topics by hypothesizing that high-frequency terms are not important for each topic. Repeated words such as "part" or "system" may not convey relevant information for all topics; therefore, the variable $\lambda$ from LDAvis is used to balance the impact of high-frequency words for a more meaningful topic interpretation. To illustrate its effectiveness, we select the top three relevant words in topic 3 of Figure 3.6 as an example. Designers may interpret topic 3 as the rolling motion of pipes based on its first seven words. With $\lambda = 1$, the original top three most relevant words are listed as "pipe," "station," and "stop" based on their probability in descent order. The words "pipe" and "station" are high-frequency words, occurring in approximately 40% and 30% of requirements, respectively, for the entire corpus. Conversely, "stop" has a relatively low overall term frequency (a smaller red and gray percentage). By setting $\lambda = 0.8$, the relevance score of "stop" bypassed "station," meaning the word "stop" can contribute more information for topic 3 and aids topic interpretation. Per domain expertise, "pipe stop" is a mechanical component commonly used to lift, roll, adjust, and transfer pipes among stations. Thus, "pipe" and "stop" should be closely related. Depending on the domain knowledge, designers should incorporate a different number of top words into each generated topic to enhance the quality of the generated topic by selecting the most suitable $\lambda$ value.

## 3.3 Generalizing Requirements into Topics

Comparing to the previous section, this segment provides a framework for generalizing design topics from requirement documents. This study has examined several model combinations to improve the word quality distributions for representing requirement topics. A data processing pipeline in Fig. 3.9 consists of four steps: (1) A data normalization step involves exploratory data analysis and preprocessing of the unstructured text into tokens. (2) With the standardized inputs, different models can learn the topic-word distributions or convert the words to vector representations as tokenized inputs. (3) A concatenation step combines two different representations as inputs for training an autoencoder. (4) The last step entails evaluating the performance of the models and visualizing the design topics.

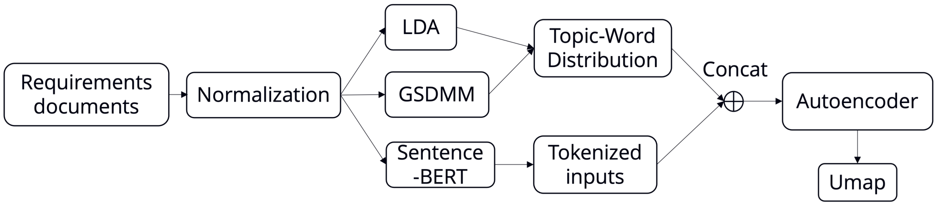


Figure 3.9: Requirements Documents Processing Pipeline

### Exploratory Data Analysis

This section provides a general framework for generalizing design topics from requirement documents using both topic modeling and word embedding. A data processing pipeline in Fig. 3.9 consists of four steps: (1) a data normalization step involves exploratory data analysis and preprocessing of the unstructured text into tokens, (2) with the standardized inputs, different models can learn the topic-word distributions or convert the words to vector representations as tokenized inputs, (3) a concatenation step combines two different representations as inputs for training an autoencoder, and (4) the final step entails evaluating the performance of the models and visualizing the design topics.

## Phase I: Text Normalization

Exploratory data analysis (EDA) is a common preprocessing practice for visualizing the main characteristics of datasets. An EDA involves identifying potential anomalies, determining the correlations among features, reducing noise, and determining the appropriate pre-processing steps for the data. Utilizing three in-house industrial datasets (Hein et al., 2018; B. W. Morkos, 2012), this study first applies EDA to visualize the distribution of words in each project's dataset. In addition to differences in design, each project has a different word count, containing 350, 160, and 247 sentences and 793, 806, and 1,051 unique words, respectively. In all three projects, we visualized average sentence length as a bar plot and estimated the Gaussian kernel density over the histogram.

To improve the performance of topic modeling, a text preprocessing step is used to identify recurring patterns of words within the text (Schofield et al., 2017). Several steps are performed using the NLTK Python library[14], including, for example: (1) spelling check, tokenization, and lowercase operation. (2) both stopword removal and lemmatization. These procedures decrease the number

[14] https://www.nltk.org

of inflectional forms for each word and allow the subsequent model to focus on more meaningful words. Typically, topic modeling with text preprocessing yields better results for topic modeling.

## Phase II: Topic Modeling

Two types of model analysis are provided by the design framework to generate design topics using three industrial requirements documents. First, we implement topic modeling, such as LDA and GSDMM. Secondly, we implement mixed models, LDA_BERT and GSDMM_BERT, which we evaluate and compare with the topic models. Using Sentence-BERT alongside topic modeling provides additional semantic and syntactic information to generalize requirements documents into distinguishable topics and to enhance topics' interpretability.

Following the text preprocessing step, LDA is implemented using the Gensim package[15]. To solve Equation 3.2, Gibbs sampling method is used to determine the posterior distribution for a total of $M$ documents and $k$ topics. The optimal number of topics for each project was predetermined in our previous study (C. Chen et al., 2021). Both $\alpha$ and $\beta$ are Dirichlet parameters. Each iteration, preconditioned on the topic probability $z_{dn}$, will compute and adjust the new topic based on the probability distributions of the word-topic matrix $\phi_{nk}$ and the document-topic matrix $\theta_{dk}$ for every word $w_{dn}$. A coherence metric is used to evaluate the performance of the trained model, which is then judged by domain experts. As compared to perplexity values, the coherence score provides more meaningful interpretations for topic modeling (Röder et al., 2015). Among the different coherence measurements, *C_v* is based on the cosine measure, normalized pointwise mutual information

[15] https://radimrehurek.com/gensim/models/ldamodel.html

(NPMI), and boolean sliding window. It ranges between 0 and 1. For requirement documents, the higher the value, the more likely the result is to be in accordance with human's judgment.

$$p(\mathcal{D}|\alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left( \prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn},\beta) \right) d\theta_d \tag{3.2}$$

Similarly, the goal of GSDMM estimates the posterior probability of short documents for each topic cluster in Equation 3.3 (Yin and Wang, 2014). In contrast to LDA, the GSDMM model assumes that each document contains only one topic. For a given $K$ number of topic clusters, Dirichlet multinomial mixture (DMM) applies the Naive Bayes assumption, which holds that the probability of each word occurring within a document is independent. The topic mixture components follow a multinomial distribution over words, $p(z = k) = \phi_{kd}$. The predefined $K$ number of topics serves as an upper limit. As a result, assigning a large number of clusters may result in empty clusters.

$$p(\mathcal{D}) = \sum_{k=1}^{K} \prod_{w \in d} p(w|z = k) p(z = k) \tag{3.3}$$

## Phase III: Concatenation

Concatenation tensors are commonly used in machine learning to represent features jointly. This study combines the output parameters from the LDA and the sentence-BERT models through a hyperparameter, $\lambda$, as a weight. For instance, in project 1, the LDA word-by-topic matrix has a dimension of 15 by 793, and the BERT embedding vector has a dimension of 793 by 768, $X_{793\times783} = \lambda \cdot X_{793\times15} \oplus (1-\lambda) \cdot X_{793\times768}$. As a result, the joint representation includes both topic-word correlations and word embedding information.

When selecting LDA_BERT or GSDMM_BERT options, an autoencoder trick is implemented to generate encoding variables, which allows the model to compress highly dimensional features into a lower dimensional space. The ADAM optimizer is coupled with mean squared error during the training process. After training and predicting on $X_{793 \times 783}$, the model learns the hidden representations of topic word distributions. UMAP (McInnes et al., 2018) is used to visualize word-topic correlations as a plot. In comparison to other dimension reduction techniques, UMAP retains both global and local structure in terms of inter-cluster relationships. Based on the learned labels, the plot can better visualize each design topic by calculating the top two eigenvalues.

Overall model performance is determined based on coherence and Silhouette scores. The Coherence score is directly evaluated for topic modeling, and the Silhouette score is used to evaluate the quality of created topic clusters (Lovmar et al., 2005). Silhouette score rates each design topic on a scale of -1 to 1. A value close to zero indicates that each data point has the same probability of belonging to other clusters. Silhouette scores that are negative indicate that a datapoint is closer to its neighbor cluster than its own cluster. The higher value represents a better graphical representation in which the average intra-cluster distance of a data point is smaller than its inter-cluster distance. For each word in a topic cluster ($i \in C_I$), Equation 3.4 computes the average distance between the word, $i$, with any other words, $j$, in the cluster, where $d(i, j)$ represents the Euclidean distance between any two-word pairs. However, Equation 3.5 calculates the intra-cluster mean distance between any word, $j$, from cluster $C_I$

to the other clusters $C_J$, where $C_J \neq C_I$. Equation 3.6 measures how similar a word is to its own cluster compared to other clusters. We have implemented the Sklearn library of Silhouette scores for the average value of all samples.

$$a(i) = \frac{1}{|C_I - 1|} \sum_{j \in C_I, i \neq j} d(i, j) \tag{3.4}$$

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j) \tag{3.5}$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3.6}$$

## Phase IV: Data Visualization

Heatmap plot is used to visualize both the LDA and GSDMM models. The goal of heatmap plots is to determine the correlation distance[16] between vocabulary and latent topics. Topics and words are then clustered hierarchically using Euclidean distance in a subspace.

UMAP is a dimension reduction technique that enables the detection of the topological structure of data by computing the top eigenvalues. Comparatively to other PCA embedding methods (such as T-SNE), UMAP can distinguish between different clusters of words based on their correlations. The color scale represents the different clustering labels. In each project, the autoencoder output is directly input into UMAP for visualizing cluster word correlations.

[16] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html

## Results and Discussion

In this section, we first show the sentence length distribution for each project in Fig. 3.10. Even though each industrial project was developed independently, their average sentence length follows a similar pattern and falls within a narrow range of 25 words. Few studies have explored how short text topic modeling can be used to efficiently generalize requirements documents. As a result, we first evaluate the model's performance by using the GSDMM to create topic-word correlations.

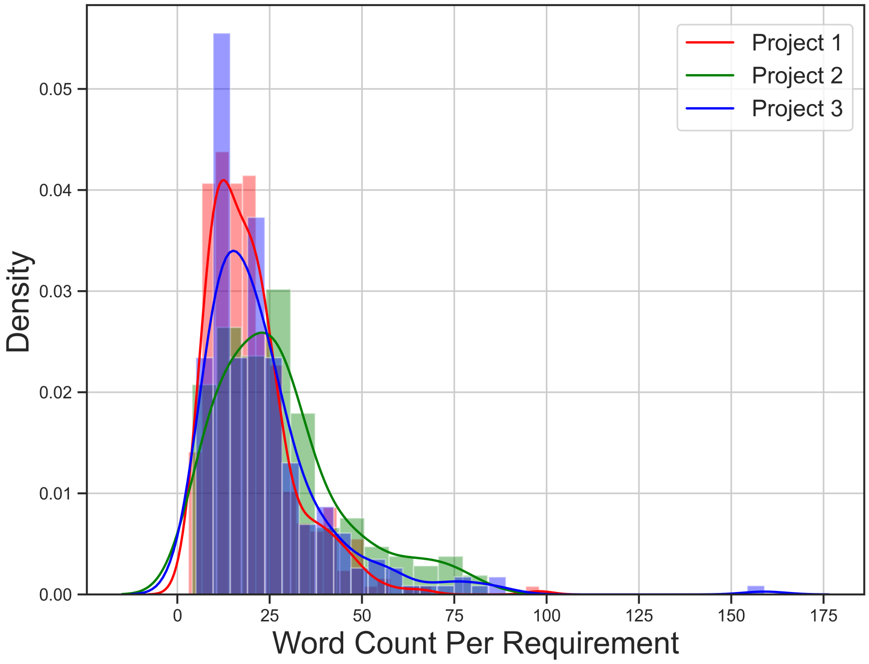


Figure 3.10: Histogram Of Word Counts For Each Project

A topic-word correlation can be identified using either LDA or GSDMM. First, we generate a hierarchically clustered heatmap representation using the Seaborn package. Each topic is manually interpreted and assigned a label. In Fig. 3.11, the legend determines the degree of semantic similarity between words and topics. There are two interesting findings regarding the correlations. The first finding allows designers to prioritize EC propagation at the topic level. For instance, the top related keywords in topic 10, such as “pipe,” “stop,” “position,” “project,” and “adjustable,” can be interpreted as relating to the motion of pipes. The pipe stop is an object available at every station to stop the rolling motion. During a redesign process for replacing pipe stops, designers could pinpoint such keywords within each requirement sentence for tracing engineering changes and verifying engineering change propagation paths. Then designers could use hierarchical order in the heatmap to determine which adjacent topics are closely related to a given topic. Consequently, each EC within topic 10 could propagate to topics 1, 3, 7, 9, and 12. According to Fig. 3.11, topic 10 is highly correlated with topics of 1, 3, 7, 9, and 12, because these topics are also related to pipe processing.

The second finding occurs on the word level, where the color scale represents the relevance within a topic. Words with a darker scale occurring within a topic suggest a closer word-topic correlation. This can be useful for tracking the change propagation of a specific component within a complex system. Tracking a keyword as a starting point enables engineers to narrow down the most related nouns and verbs strongly associated with the component. For instance, based on the hierarchy diagram, the highest frequency word related to “welded” is “tube” in topic 7. On the higher level, the term, Welded Tube, is part of the name of a company and often correlated to “installation” and “startup.” Based on the interpretation, the company has a responsibility to install and configure certain equipment for this project. On the topic level, topic 7 is directly linked to topic 9, where any EC can lead to other responsibilities for the company, including the adjustment of “threading

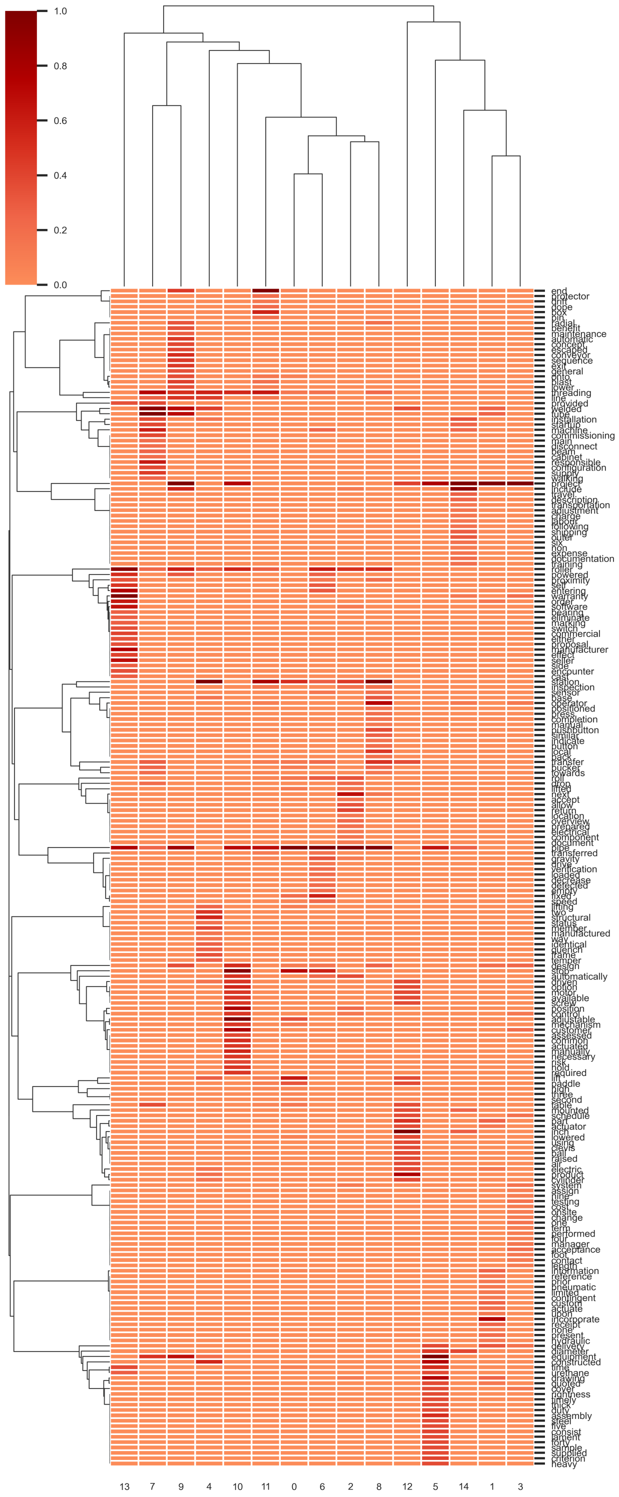


Figure 3.11: Topics And Words Distribution From LDA For Project 1

line" or "conveyor" systems. Note that most functional requirements are closely associated with high relevance words, such as "pipe." In contrast, the non-functional requirements contain diverse and low-frequency vocabulary, where the keywords are often displayed with a brighter color scale.

To enhance topic modeling performance, a pre-trained sentence-BERT model is coupled to topic modeling to incorporate the general knowledge for representing design topics. As topic models are evaluated by coherence score, Silhouette score measures the topic clustering performance. Table 3.7 shows the model performance among LDA, GSDMM, LDA_BERT, and GSDMM_BERT compared for all three projects. Each value is averaged over five runs, and the highest scores are highlighted in bold. After combining the sentence-BERT model with topic models, each project improves their model performance to various degrees. In general, both LDA_BERT and GSDMM_BERT outperform the topic modeling for better representing design topics.

Table 3.7: Model Performance With Industrial Projects (averaged over 5 runs)

| | Project 1 | | Project 2 | | Project 3 | |
|---|---|---|---|---|---|---|
| | Coherence | Silhouette | Coherence | Silhouette | Coherence | Silhouette |
| *LDA* | 0.4124 | - | 0.4156 | - | 0.4001 | - |
| *GSDMM* | 0.4751 | - | 0.5346 | - | 0.3856 | - |
| *LDA_BERT* | 0.5000 | 0.2881 | **0.5579** | 0.3730 | **0.4272** | **0.3812** |
| *GSDMM_BERT* | **0.5480** | **0.3987** | 0.5327 | **0.3766** | 0.3716 | 0.3538 |

In Table 3.8, the topic identified as threading stations are selected and compared across different models. For each method, top ten keywords are selected for evaluating the quality of the word distribution. Though certain keywords are somehow related to threading stations, they are not related to one another in this case. Such words are highlighted in red as they are considered irreverent. For instance, LDA contains words, such as "pin" and "dope," that have no semantic contribution to the topic. In comparison with LDA, LDA_BERT elevates more relevant words, such as "pipe," and excludes "dope," to improve topic qualities. In addition to LDA-based models, GSDMM and GSDMM_BERT can also provide competitive results by capturing slightly different aspects of pipe processing keywords. By incorporating the sentence-BERT model, the pre-trained syntactic relations between words can enhance the quality of each generated topic.

Table 3.8: Different Models' Word Distributions Of Pipe Threading Stations In Project 1

| **Models** | **Top 10 words** |
|---|---|
| *LDA* | threading, end, station, line, box, pin, inspection, protector, dope, drift |
| *GSDMM* | pipe, end, station, threading, box, stop, lift, roller, paddle, inspection |
| *LDA_BERT* | station, end, threading, pipe, box, roller, line, protector, stop, drift |
| *GSDMM_BERT* | pipe, station, stop, end, threading, fixed, box, adjustable, roll, gravity |

As multiple FRs and non-FRs can refer to the same topic, tracing the topic to requirements can be difficult. To visualize the word distributions for each topic, Figures 3.12 and 3.13 project words in a lower-dimensional space. Each dot represents a unique word, and each word is assigned to one of the topics with a distinct color. Each topic has a normalized percentage value representing the ratio of its unique words to the entire vocabulary in Project 1. In both figures, a sampled topic is selected for further discussion and comparison. As the main theme of Project 1 pertains to the general pipe manufacturing process, most of the topics are adjacent to topic 1 in the subspace. The topic 1 in Figure 3.12 represents the threading pipe station, which has many keywords associated with the functions of pipe stations. Topic 3 in Figure 3.13 focuses on the details of project timelines, which are closely related to non-FRs requirements, including personnel training requirements and responsibilities for different pipe stations. As the results are presented via dimensional reduction techniques, the approach can efficiently handle large corpora.

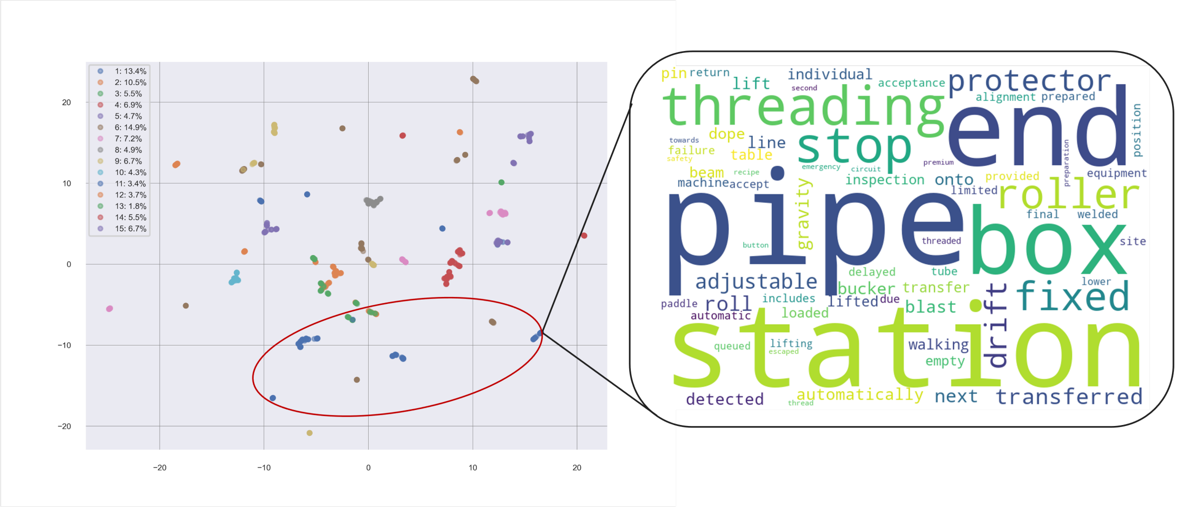


Figure 3.12: Topics And Words Distribution In Subspace With A Highlighted Topic 1 For Project 1

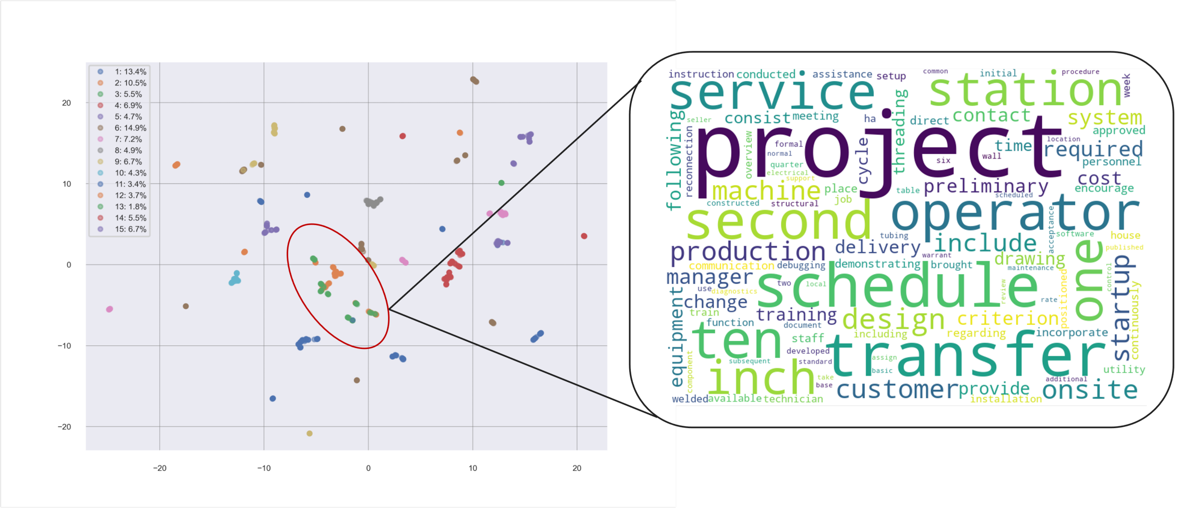


Figure 3.13: Topics And Words Distribution In Subspace With A Highlighted Topic 3 For Project 1

From the perspective of engineering practices, this approach implements a formal data pipeline to determine the design topic from the requirements documents. Depending on the size of documents and purpose, designers can choose an appropriate method to generalize design topics with a low computational cost. In topic modeling, a heat map plot shows the EC based on the semantic correlations among words for a smaller size design requirements document. When dealing with complex system requirements documents, utilizing topic modeling and word embedding can help visualize and interpret intrinsic requirements correlations in a subspace. In accordance with domain knowledge, initial requirements change to a mechanical component should verify keywords within each topic. A subsequent change may propagate to adjacent topics via different keywords, influencing the actions, functions, characteristics, and behaviors of other components. Through the generation of hidden topics, our proposed framework provides designers with a means to better understand the structure of requirements for complex designs and interpret the corresponding EC propagation.

There are two major conclusions to be drawn from this study. (1) This research provides insight into how both topic modeling and word embedding models can be used to improve the quality of requirements design topics. This study suggests that a combined model can better extract topics from industrial requirements documents and provide better model performance and higher quality word distributions than LDA alone. (2) With predetermined topics to narrow the scope of design changes, engineers could quickly identify related requirements. Upon further development, this work can be integrated into commercial requirement management software in smart manufacturing.

## Limitations

As discussed above, both LDA_BERT and GSDMM_BERT have several critical limitations:(1) Automatically determining the number of topics still need to be explored. The current data pipeline can be combined with other models to obtain the optimal result; (2) As many of the model hyperparameters are determined manually, a tuning procedure is required for finding the best values while minimizing computational cost; (3) The entire process of tracking engineering changes using requirement topics is not fully automated. After addressing these issues, these results can then be combined with downstream analysis to build correlations with geometry modeling.

## 3.4 Point Cloud Classification

This section focuses on how we can recognize mechanical subassembly designs from given CAD models. To improve recognition of engineering changes (ECs) in terms of mechanical components, this framework focuses on developing an algorithm to classify CAD models in terms of point clouds into predefined categories. Identifying the quality of mechanical design automatically could lead to lean manufacturing in practice. Using this proposed model, engineers can identify, correct, and verify the qualities of mechanical components for various applications.

### Meta-SeL

**Data**. In the absence of an online benchmark dataset for the field of design and manufacturing, this study utilizes a subset of ShapeNetCore datasets to represent mechanical CAD designs. We implement a preprocessing step to filter out CAD models with fewer than or more than three parts (e.g., subassemblies). After narrowing the datasets from 17,775 to 7,555, we divide the filtered models into 90% for training and 10% for testing. As shown in Tables 3.10 and 3.11, the 16 categories are reduced to 10 and a detailed breakdown of each category is provided. There is only one model left in the motorbike category, which has been manually removed. The following calculations are based on filtered CAD models. There are several techniques for injecting noise into point clouds, including normalizing the data, randomly rotating for one of the axes (x, y, z), translating coordinates randomly, and jittering the points. Individual and collective tests of such procedures are conducted during the training phase to improve the generalization performance for testing datasets. The best combination is presented and discussed in the results section.

Table 3.9: Comparison of ShapeNetCore Datasets After Filtering (Number of Parts = 3)

| | # of Samples | # of Filtered Samples |
|---|---|---|
| Training: | 15,990 | 6,805 |
| Testing: | 1,785 | 749 |
| Total: | 17,775 | 7,555 |

Table 3.10: Breakdown of Training Sets By Each Category

| Categories: | Labels: | # of Models |
|---|---|---|
| Airplane: | 0 | 471 |
| Car: | 3 | 257 |
| Chair: | 4 | 2,508 |
| Earphone: | 5 | 31 |
| Guitar: | 6 | 706 |
| Lamp: | 8 | 1,086 |
| Pistol: | 12 | 244 |
| Rocket: | 13 | 51 |
| Skateboard: | 14 | 102 |
| Table: | 15 | 1,349 |
| Total: | | 6,805 |

**Training**. The MetaSeL algorithm is comprised of two major components, SAE and Meta-learning techniques, as shown in Figure 3.14. For each model, the SAE is calculated first to learn their semantic representation. MATLAB is used to implement CPU parallel processing to speed up the training process. Using the Sylvester equation, a 3-by-3 latent

Table 3.11: Breakdown of Testing Sets By Each Category

| Categories: | Labels: | # of Models |
|---|---|---|
| Airplane: | 0 | 51 |
| Car: | 3 | 31 |
| Chair: | 4 | 281 |
| Earphone: | 5 | 2 |
| Guitar: | 6 | 79 |
| Lamp: | 8 | 115 |
| Pistol: | 12 | 28 |
| Rocket: | 13 | 6 |
| Skateboard: | 14 | 11 |
| Table: | 15 | 145 |
| Total: | | 749 |

matrix in semantic space is calculated and solved using the Bartels-Stewart algorithm in Equation (3.7, 3.8). An example of a single model is shown in Figure 3.15 which consists of 1024 points with coordinates (x, y, z) on the left-hand side. Right hand side, 3 signifies that each point has a label that belongs to one of the three pre-assigned parts. Figure 3.16 illustrates how we iterate this process by computing the weights for each model,

$$\underbrace{SS^T}_{A} W + \underbrace{\lambda X X^T}_{B} W = \underbrace{(1+\lambda) S X^T}_{C} \tag{3.7}$$

where $X \in \mathbb{R}^{d \times N}$ is the input data with $N$ feature vectors and $d$ dimensions. $S \in \mathbb{R}^{k \times N}$ indicates the latent representation of a linear autoencoder. $W \in \mathbb{R}^{k \times d}$ represents the projection matrix while $k < d$.

$$AW + BW = C \tag{3.8}$$

**Testing**. For each test model, we use the same procedure to determine the SAE first, as shown in Figure 3.17. We compare the cosine similarity between the weights between the test and training sets to determine the object category

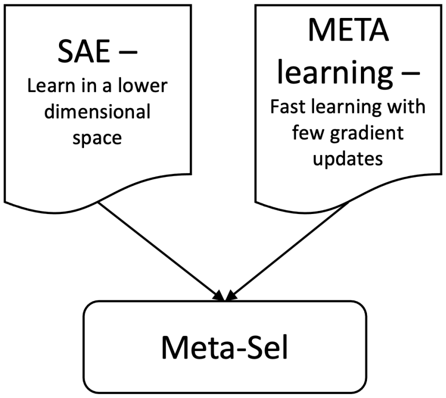

SAE –
Learn in a lower dimensional space
META learning –
Fast learning with few gradient updates
Meta-Sel

Figure 3.14: MetaSel Model System Level Component

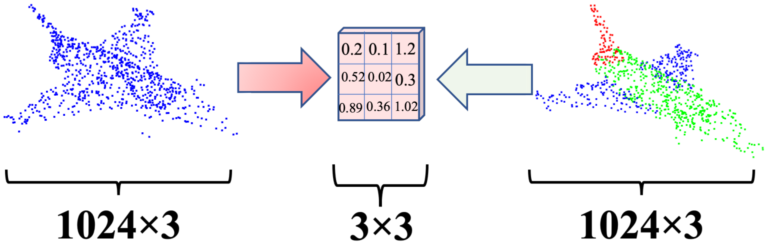


Figure 3.15: A Conceptual Representation Of Calculating SAE For Each Cad Model

for testing models. As a result, we select the model with the highest probability as its label. The classification accuracy is calculated by comparing the most likely label with the ground truth.

## Results and Discussion

By using random shuffling (RS) in Table 3.12, we demonstrate that our method is permutation invariant for the order of models. The result will not be affected by random shuffling of datasets. We then demonstrate that normalizing each model into a unit sphere to improve classification efficiency. In accordance with the normalization procedure (N), a random rotation (RR) is performed on one of the three axes (x, y, z) but the results did not provide a significant improvement. As a next step, we test translation, jittering, and combinations of these methods. In translation, noise is generated by drawing

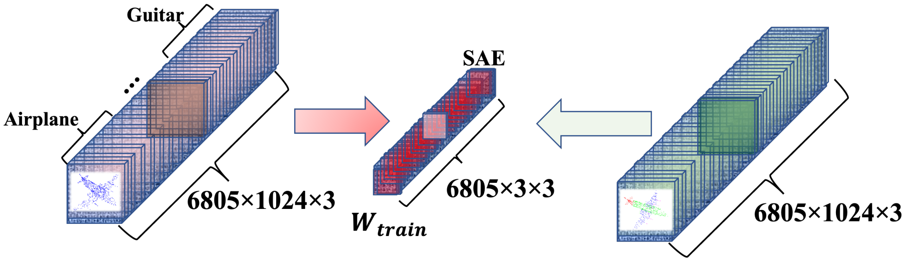


Figure 3.16: Model Architecture Of MetaSel During Training

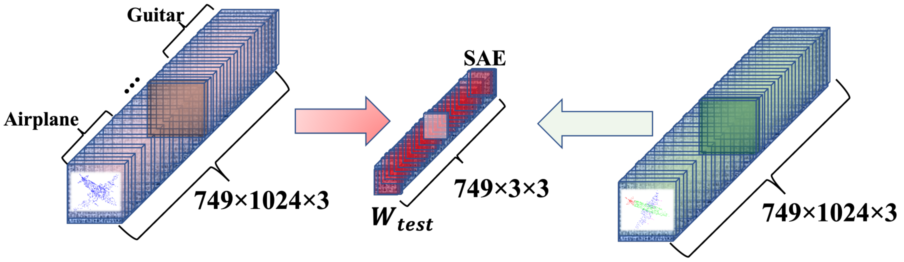


Figure 3.17: Model Architecture of MetaSel During Testing

new samples at uniform intervals based on uniformly distributed samples. As a result of jittering, each coordinate is subjected to a Gaussian noise with a zero mean and a standard deviation of 0.01. In bald, the best results are highlighted as the final recommendation.

| Meta-SeL: | Accuracy (%) |
|---|---|
| Base : | 93.19 |
| Random Shuffle (RS) | 93.19 |
| $Normalization(N)$ | **95.59** |
| $N + RR(x - axis)$ | 90.25 |
| $N + RR(y - axis)$ | 93.59 |
| $N + RR(z - axis)$ | 83.97 |
| $N + RR(x, y, z)$ | 88.38 |
| $N + T$ | **95.99** |
| $N + RR + T$ | 77.03 |
| $N + J$ | **95.95** |
| $N + J + T$ | **95.46** |

Table 3.12: Comparison of Meta-SeL Results with Various Noise Techniques

Meta-SeL's base model with a normalization preprocessing step was selected as one of the best results for determining accuracy for each category. The accuracy of each category, as well as the average recall and precision, are shown in Table 3.13. Skateboards and earphones have the lowest accuracy across categories. Skateboards have the highest misclassification rate, with seven of them being classified as lamps. The results of the study show that certain CAD objects may share certain characteristics, causing a misclassification error. To investigate this further, we project all the training weights into latent space using UMAP to compare their characteristics. In Figure 3.18, all 6,805 SAE weights (e.g., 3-by-3 matrix) are projected in a subspace labeled based on the number of categories. Ideally, objects within the same category would be more similar to each other than they would be to features belonging to other clusters. However, certain models have similarities between categories, which can lead to misclassification.

Table 3.13: Comparison of Predicted Results by Categories

| Categories: | Training: | Testing: | Recall (%) | Precision (%) |
|---|---|---|---|---|
| Airplane: | 471 | 51 | 94.11 | 96 |
| Car: | 257 | 31 | 93.33 | 96.55 |
| Chair: | 2508 | 281 | 99.64 | 98.93 |
| Earphone: | 31 | 2 | 50 | 25 |
| Guitar: | 706 | 79 | 98.73 | 98.73 |
| Lamp: | 1086 | 115 | 91.30 | 92.10 |
| Pistol: | 244 | 28 | 100 | 90.32 |
| Rocket: | 51 | 6 | 100 | 66.66 |
| Skateboard: | 102 | 11 | 36.66 | 66.66 |
| Table: | 1349 | 145 | 96.55 | 97.90 |
| Total/Avg Result: | 6,805 | 749 | 95.99 | 96.12 |

Meta-SeL's performance is demonstrated by comparing it to Pointnet and DGCNN algorithms on the same datasets and setups. Figure 3.19 shows that our model provides a competitive level of accuracy. Our model calculates SAE and cosine similarity simultaneously, resulting in high initial accuracy. Using a variety of input datasets and different techniques, we demonstrate that the

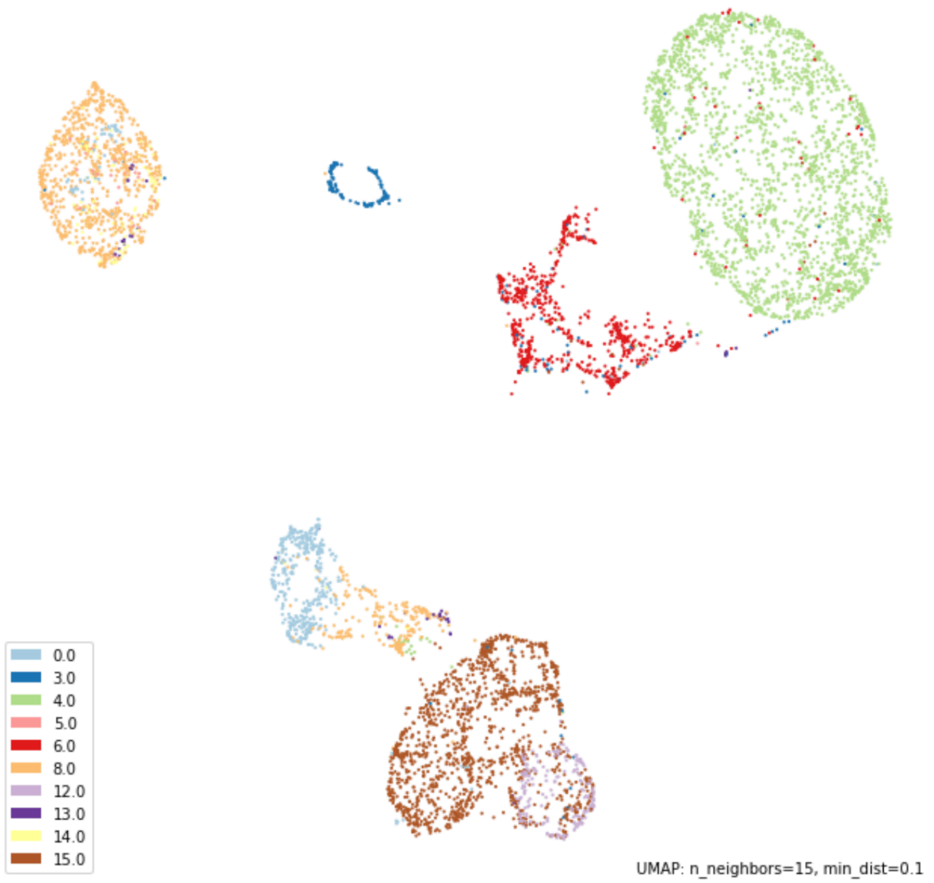


Figure 3.18: Projection of Training Weights into Latent Space Using UMAP

deep learning model will eventually bypass Meta-SeL and achieve a higher level of accuracy.

**Contributions**. We argue that Meta-SeL provides competitive results with certain state-of-the-art models. The major contributions are summarized as follows:

- Our model can reduce training time and provide high accuracy predictions when new CAD models are added to the dataset.

- Meta-SeL is permutation-invariant for the order of models and can produce consistent predictions.

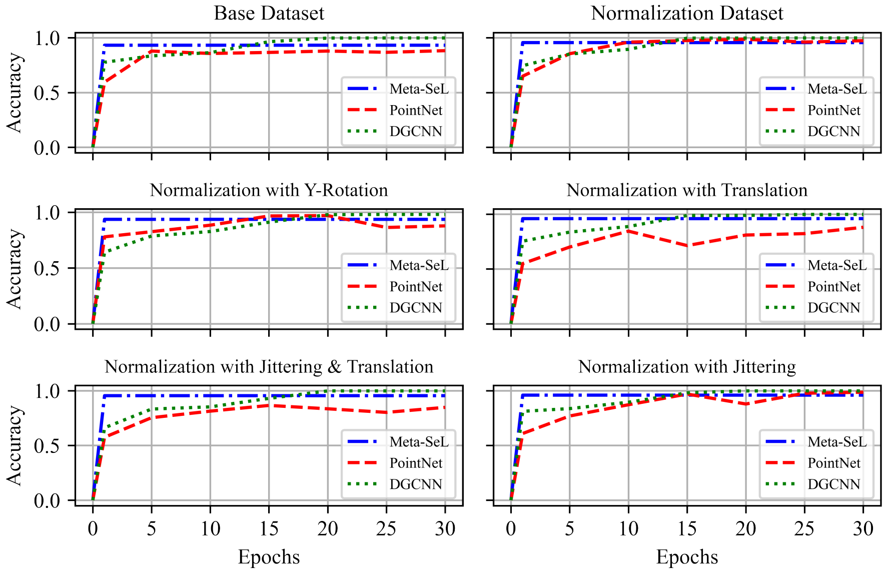


Figure 3.19: Result Comparison of Meta-SeL Other Model Architectures

- Meta-SeL is sensitive to model input, and certain noise injection techniques can improve its performance.

**Limitations** There are several limitations to the current version of the model. (1). Our approach is capable of handling models with three parts at present, thus different number of components of point cloud representations should be explored in the future. (2). It is necessary to perform different preprocessing treatments to distinguish the categories with similar geometry characters and further improve the accuracy of classification. (3) Although our shallow learning algorithm is effective in classifying each category, other model architectures should also be developed to further reduce memory requirements.

## 3.5 Linking Requirements to CAD Images

In this section, we present a framework for recognizing the relationships between images and requirements. To represent requirements documents' physical components, we generate a synthetic image dataset from an online database based on our in-house requirements document gathered from industry. Figure 3.20 shows the pipeline of the proposed framework using a fine-tuned CLIP model.

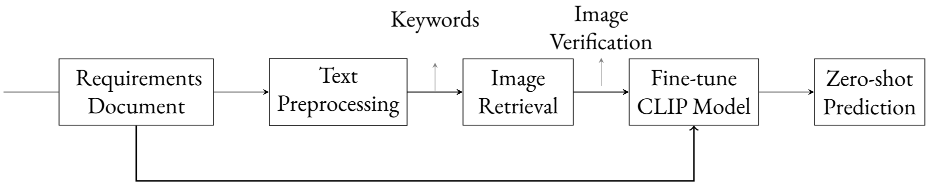


Figure 3.20: Pipeline Of Proposed Framework

### Text Preprocessing

The purpose of a text preprocessing step is to extract the most relevant information and use it as keywords for image scraping, because particular words contribute more to connecting visual ideas than others. Our first step is to eliminate all non-alphanumeric characters and stopwords (e.g., "shall," "etc.," or "must"). The remainder of the corpus consists of nouns, verbs, and adjectives filtered by part-of-speech (POS) tagging. A previous study determined that nouns and verbs can be used to describe the physical architecture and functional characteristics of projects (Hein et al., 2018). In this study, adjectives were also considered relevant to describe these components. These keywords are stored and applied to search queries.

## Synthetic Image Dataset

An industrial requirement dataset describing the design of a pipe assembly line is implemented in this study (B. W. Morkos, 2012). In total, 350 requirements are included, containing both functional and non-functional requirements. Project details include topics such as design specifications, project descriptions, equipment supplies, installation procedures, and shipping. After text preprocessing, each sentence is reduced to phrases for retrieving online images.

A version of the search model is implemented to scrape images from online text searches. As the order of keywords does not significantly affect the search results, queries are automatically sent to online servers to retrieve images as a browser user. BeautifulSoup, Request, LXML XML toolkit, and regular expressions are used to get image links and download the original resolution image locally. As images can be extracted from several sources, a verification procedure is implemented to ensure that all images are accessible through the Pillow library. For example, some images cannot be downloaded from online PDF documents or websites protected by anti-bot tools such as CAPTCHA. This requires manual verification to replace irrelevant images. Because images come in a variety of sizes, we use the resampling LANCZOS[1]filter to rescale each image into a 300 × 300 pixel size. By doing so, we avoid losing information on the edges.

## CLIP Model

As the number of image-requirement pairs is relatively small, directly training the model on CLIP might not be effective. Instead, transfer

[17] https://pillow.readthedocs.io/en/stable/releasenotes/2.7.0.html

learning allows the model to integrate previous knowledge with domain-specific knowledge. In this experiment, we compared the performance of pre-trained and fine-tuned CLIP models. Conducting an overall evaluation of zero-shot prediction accuracy is beyond the scope of this study.

### Prediction on Pre-trained CLIP

Using a pre-trained CLIP model, we select an image closely associated with the industrial design to predict the most likely requirements from the existing design document. Before passing to the image encoder, the new image must go through the same filters. The transformer model is used to encode requirements. By utilizing zero-shot predictions, the most relevant requirements are identified. As the pre-trained CLIP model is trained to perform general tasks, a fine-tuned prediction model should provide improved performance when applied to domain-specific knowledge.

### Prediction on Fine-tuned CLIP

The requirement-image pair is first randomly shuffled into a training set with a batch size of ten. Testing the model involves implementing zero-shot prediction, in which out-of-distribution images are manually downloaded from variant designs. The total number of epochs is twenty. Image and text losses are calculated individually using cross-entropy. The Adam optimizer is implemented with a learning rate of 5e-6 and decoupled weight decay regularization of 0.4 for all layers. These values are adjusted based on analysis and evaluation to fine-tune the hyperparameters. Similarly, the same prediction procedure is implemented to output the top five requirements with their probabilities.

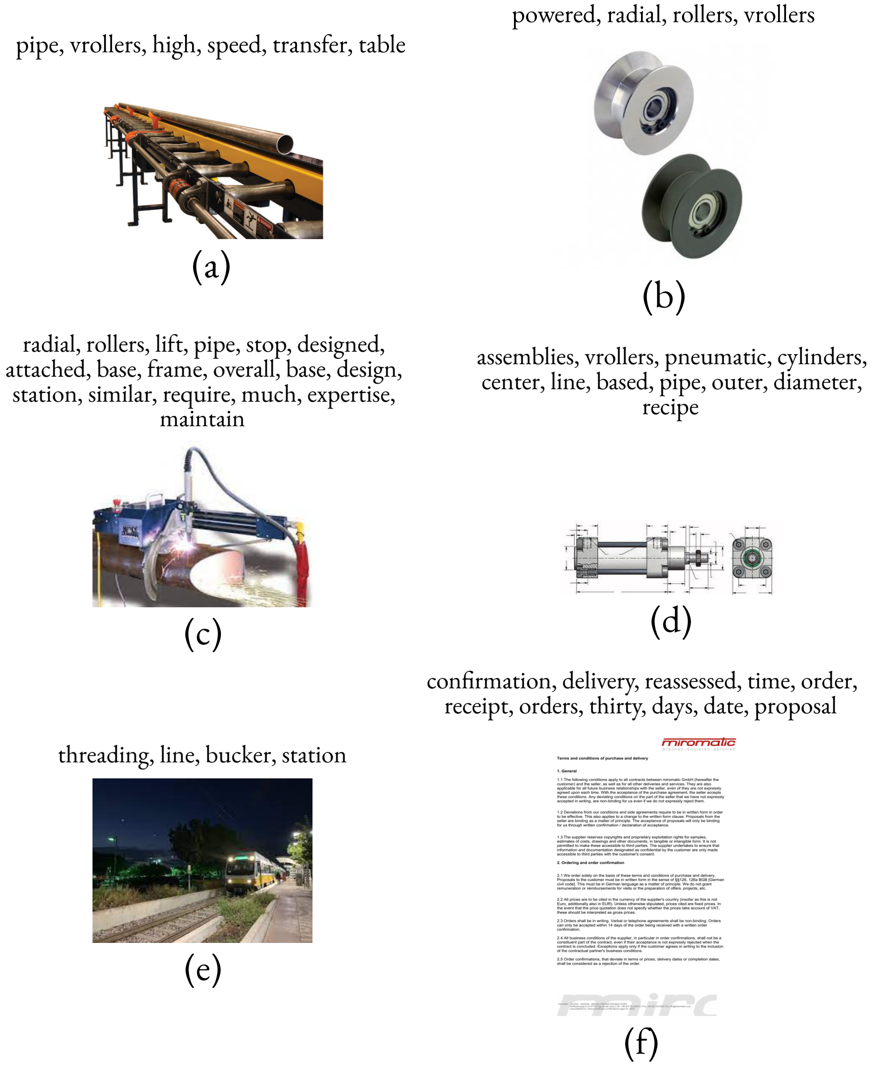
## RESULT AND DISCUSSION

Study findings revealed interesting observations that could help bridge the gap between requirements and images. Observations relating to the type, quality, and relevance of predicted requirements are discussed. These results will demonstrate the improvement in the fine-tuned model with its interpretable results.

### Synthetic Dataset

As the created synthetic image dataset contains various types of images, Figure 3.21 presents several search results from the industrial trial project. In response to different search terms, the collected images include photographs, drawings, and document scans. Note that not all the returned images accurately reflect the details of a search query, and we assume the top images are the most relevant ones. If the first image is not available, the next resemblance image is downloaded manually. Further, some images may not capture the meaning of the requirements due to ambiguous words and short search queries. In such cases, we consider some images to be noise. For example, in Figure 3.21 (e), upon sending the query "threading, line, Bucker, station," the retrieved result depicts a picture of a train departing the Buckner station.

Although a fine-tuned model may not learn valuable knowledge from irrelevant images, it is still possible to obtain limited useful information. In Figure 3.21 (f), many search queries related to non-functional requirements contain the words "proposal," "description," "specification," and "criteria," which result in a screenshot of a document. Though CLIP models may not capture detailed content from images, they may still recognize these keywords as representing the concept of documents. In context-rich design projects that



pipe, vrollers, high, speed, transfer, table

(a)

powered, radial, rollers, vrollers

(b)

radial, rollers, lift, pipe, stop, designed, attached, base, frame, overall, base, design, station, similar, require, much, expertise, maintain

(c)

assemblies, vrollers, pneumatic, cylinders, center, line, based, pipe, outer, diameter, recipe

(d)

threading, line, bucker, station

(e)

confirmation, delivery, reassessed, time, order, receipt, orders, thirty, days, date, proposal

**Terms and conditions of purchase and delivery**

**1. General**

1.1 The following conditions apply to all contracts between miromatic GmbH (hereafter the customer) and the seller, as well as for all other deliveries and services. They are also applicable for all future business relationships with the seller, even of they are not expressly agreed upon each time. With the acceptance of the purchase agreement, the seller accepts these conditions. Any deviating conditions on the part of the seller that we have not expressly accepted in writing, are non-binding for us even if we do not expressly reject them.

1.2 Deviations from our conditions and side agreements require to be in written form in order to be effective. This also applies to a change to the written form clause. Proposals from the seller are binding as a matter of principle. The acceptance of proposals will only be binding for us through written confirmation / declaration of acceptance.

1.3 The supplier reserves copyrights and proprietary exploitation rights for samples, estimates of costs, drawings and other documents, in tangible or intangible form. It is not permitted to make these accessible to third parties. The supplier undertakes to ensure that information and documentation designated as confidential by the customer are only made accessible to third parties with the customer's consent.

**2. Ordering and order confirmation**

2.1 We order solely on the basis of these terms and conditions of purchase and delivery. Proposals to the customer must be in written form in the sense of §§126, 126a BGB [German civil code]. This must be in German language as a matter of principle. We do not grant remuneration or reimbursements for visits or the preparation of offers, projects, etc.

2.2 All prices are to be cited in the currency of the supplier's country (insofar as this is not Euro, additionally also in EUR). Unless otherwise stipulated, prices cited are fixed prices. In the event that the price quotation does not specify whether the prices take account of VAT, these should be interpreted as gross prices.

2.3 Orders shall be in writing. Verbal or telephone agreements shall be non-binding. Orders can only be accepted within 14 days of the order being received with a written order confirmation.

2.4 All business conditions of the supplier, in particular in order confirmations, shall not be a constituent part of the contract, even if their acceptance is not expressly rejected when the contract is concluded. Exceptions apply only if the customer agrees in writing to the inclusion of the contractual partner's business conditions.

2.5 Order confirmations, that deviate in terms or prices, delivery dates or completion dates, shall be considered as a rejection of the order.

miromatic mischen . dosieren . abfüllen | Michael Rothdach GmbH
Raiffeisenweg 5 | D-87743 Egg an der Günz | Tel. +49 (83 33) 9200-0 | Fax +49 (83 33) 9200-33 | info@miromatic.com
Geschäftsführer: Helmut Rothdach | HRB Memmingen Nr. 8854

(f)

Figure 3.21: Samples Of Collected Synthetic Image Datasets With Requirement Keywords

include more image documents, designers may fine-tune the model or combine it with additional neural networks to further extract textual information from images.

Similar search queries might return the same image. As an example, after word preprocessing, query numbers 158 (‘box’, ‘end’, ‘threading’, ‘station’, ‘idler’, ‘radial’, ‘rollers’, ‘vrollers’) and 167 (‘box’, ‘end’, ‘threading’, ‘inspection’, ‘station’, ‘idler’, ‘pipe’, ‘radial’, ‘rollers’) have the same image result. As both sentences contain many similar words and describe similar objects, the study uses the same pictures to represent both requirements.

### Improvements in Design

With the increasing number of epochs, the total loss decreases, as shown in Figure 3.22. The loss function is averaged based on the cross-entropy loss between the image and the text. As a result of model fine-tuning, training loss is significantly reduced (around 65%) after 10 epochs. As a trade-off decision, fine-tuning a model could result in the loss of transfer knowledge and the acquisition of more domain-specific information while increasing the number of epochs. Thus, we employed an early stop strategy during the fine-tuning process to prevent overfitting. The CLIP model stops learning requirement-image pairs after 20 epochs and provides the most interpretable results. It is important to recognize that fine-tuning increases the risk of losing previous knowledge and gaining excessive domain-specific knowledge.

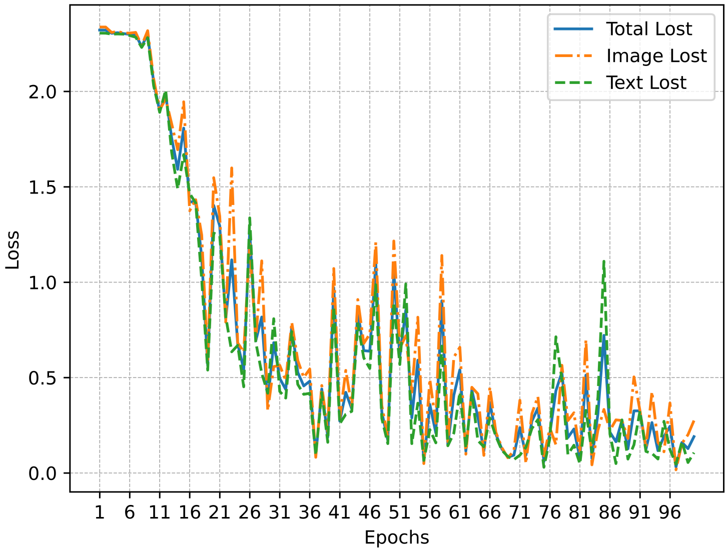


Figure 3.22: Variance Of Training Error With Increasing Epochs

The search queries, as shown in Figure 3.23, represent each requirement. A fine-tuned CLIP model is tested using an out-of-distribution image from a variant design in Figure 3.23, which shows a portion of a storage system.

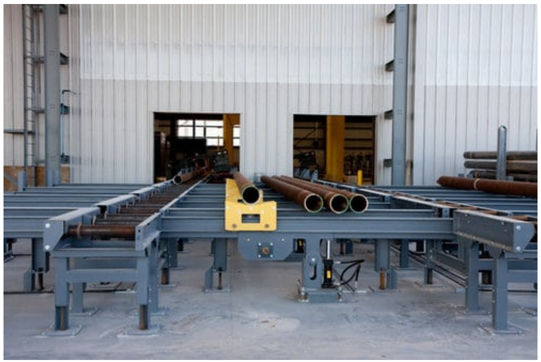
The pipe threading equipment outlined in the requirements document, as well as the storage equipment shown in Figure 3.23, contain several types of conveyor systems that can be potentially adapted from one to another.

**Validation** Rather than viewing this problem as a pure classification process, each requirement might correspond to multiple images or vice versa. The zero-shot prediction method is employed to compute the probability for each requirement-image pair. As this is an early-stage study, the focus is primarily on modeling the individual correlations rather than capturing the many-to-many relationship. Correlations of this type are not well understood and may not be sufficient to generate manually. Alternatively, designers can interpret the predicted requirements based on their intuitive understanding of their domain knowledge to the unforeseen images.

Based on the results, the best result (10.23%) is considered the most relevant requirement for the pre-trained model. In contrast, the top prediction result from the fine-tuned model achieves higher accuracy by providing more relevant information. Upon interpretation, the improved results have a closer relationship to functional requirements pertaining to "pipe stations" or "transfer tables." As the fine-tuned model can recognize the concept from images and find the most relevant requirements, engineers should determine the appropriate number of relevant requirements and make corresponding engineering adjustments.

A particularly interesting and noteworthy observation is the use of images that contain both image and text data. The image in Figure 3.24 is chosen as

[18] https://omtec.com/catalog/f1-conveyor-table/

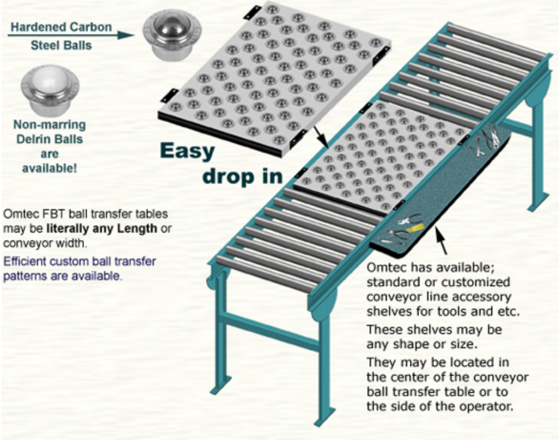


| Pre-trained Model | | Fine-tuned Model | |
|---|---|---|---|
| Keywords | Percentage | Keywords | Percentage |
| 'lift', 'pipe', 'entry', 'end', 'table', 'paddle', 'threading', 'conveyor', 'transfer', 'box' | 10.23% | 'vrollers', 'pipe', 'transfer', 'table', 'gravity', 'roll', 'towards', 'exit', 'conveyor' | 32.66% |
| 'station', 'bucker', 'structural', 'contructed', 'frame', 'members' | 5.09% | 'rail','assemblies','spaced','half','feet', 'spanning','length','transfer','table' | 13.12% |
| 'thirteen', 'table', 'line', 'threading', 'transfer' | 3.62% | 'pipe','rest','adjustable','pipe','stop', 'exit','conveyor' | 11.98% |
| 'project', 'description' | 3.59% | 'pipe','secured', 'vrollers','clamp','high', 'speed','transfer','table' | 7.31% |
| 'constructed', 'inch', 'structural', 'table', 'walls', 'tubing', 'transfer', 'quarter' | 2.89% | 'pipe','secured','vrollers','clamp','high', 'speed','transfer','table' | 7.31% |

Figure 3.23 & Table 3.14: An Image Of Conveyor System[2] With Model Predictions

a challenge for the fine-tuned model recognizing shapes and text information simultaneously. The image depicts a conveyor ball transfer table, on which hardened carbon steel balls are used to replace rollers. In such images, the fine-tuned CLIP model did not result in significant performance improvements for the top prediction (2.5% improvement), as shown in Figure 3.24. In images that contained only photographic images and no text, the fine-tuned CLIP model demonstrated superior performance (22.4% improvement).

In the pre-trained model of Figure 3.24, two distinct requirements resulted in the same keyword phrases after the pre-processing step. Both pre-trained and fine-tuned models can recognize the new image as a type of transfer table based on the given functional requirements. Upon interpretation, the predicted requirements from the fine-tuned model are more closely related to the transfer table and its functionality.


Hardened Carbon Steel Balls
Non-marring Delrin Balls are available!
Easy drop in
Omtec FBT ball transfer tables may be literally any Length or conveyor width.
Efficient custom ball transfer patterns are available.
Omtec has available; standard or customized conveyor line accessory shelves for tools and etc.
These shelves may be any shape or size.
They may be located in the center of the conveyor ball transfer table or to the side of the operator.

| Pre-trained Model | | Fine-tuned Model | |
|---|---|---|---|
| Keywords | Percentage | Keywords | Percentage |
| 'threading', 'line', 'thirteen', 'transfer', 'table' | 15.76% | 'pipe', 'pin', 'threading', 'station', 'transfer', 'table', 'towards', 'end', 'threading', 'inspection', | 18.03% |
| 'pipe', 'next', 'transfer', 'table' | 7.65% | 'design', 'vrollers', 'many', 'similar', 'features', 'vrollers', 'tube', 'uses', 'exception', 'high', 'temperature', 'designs' | 14.06% |
| 'pipe', 'next', 'transfer', 'table' | 7.65% | 'threading', 'station', 'base', 'design', 'similar', 'stations' | 9.63% |
| 'pipe', 'gravity', 'roll', 'transfer', 'table', 'towards', 'box', 'drift', 'threading', 'protector', 'station' | 4.87% | 'pipe', 'gravity', 'roll', 'transfer', 'table', 'towards', 'bucker', 'station' | 5.83% |
| 'transfer', 'table', 'designed', 'located', 'previous', 'next', 'operation' | 3.41% | 'pin', 'end', 'blast', 'station', 'design', 'identical', 'box', 'blast', | 5.45% |

Figure 3.24 & Table 3.15: An Image Of Conveyor Ball Transfer Table[3] With Model Predictions

The out-of-distribution images are selected from a variant design as indicated earlier. As similarities can be defined from different perspectives, the out-of-distribution images may take different forms. For instance, taking images of the same object from various angles with a variety of backgrounds may also be considered as testing images. As not all the mechanical components are symmetrical, different angles of the same part might have an impact on the predictions.

The study suggests that the proposed framework could potentially be used to visualize requirement traceability by taking images of various physical components. The most relevant requirements should be determined for each image and evaluated regarding engineering changes. Although the synthetic dataset contains some irrelevant images as noise, the fine-tuned CLIP model is still capable of learning useful information and improving out-of-distribution prediction.

Through a synthetic dataset, the fine-tuned model can identify standard mechanical components from collected images. For specialized mechanical parts, the image obtained from the internet may not accurately reflect their physical components. A minor change in design may, however, be treated by an out-of-distribution prediction and not necessitate a new simulation. As requirements are often added or deleted during the reengineering process, designers need to repeat the analysis to achieve higher accuracy. The proposed process would allow engineers to realize the interconnection of heterogeneous data quickly and reduce human error in the design process. Future work should explore different rotation-invariant techniques to build a more robust model and integrate this framework into digital threads. Rather than using 2D images, 3D point clouds could be another future direction. Further, the fine-tuned model can be combined with augmented reality for industrial applications.

**Limitations** The framework has the following limitations. (1). Currently, we are implementing both image- and text-preservation encoders; however, other data combinations (e.g., 3D models) should be explored for a more comprehensive evaluation. (2). A further analysis should be conducted to improve the model's performance by comparing different fine-tuning techniques and loss functions. These limitations may have an impact on the accuracy of the zero-shot learning outputs, as well as the interpretation of the results.

# Chapter 4

# Conclusion

**Requirement Topics**. The motivation of this dissertation is to build a framework for engineers to manage the complex MBE system by identifying engineering changes within downstream and upstream components. A key component of managing engineering changes is the tracking of information across different domains and data types. As part of smart manufacturing, digital threads support the industry's need to integrate information flow and provide interoperability for a variety of data. To support such an information management system, we have narrowed our study scope between the requirements and CAD domains. The study can be divided into three parts: requirement management, CAD, and requirement-to-CAD.

The challenge in the requirement domain is to track information through highly domain-specific knowledge documents. As the preliminary study utilized LSA to analyze two industrial projects, it is possible to cluster the requirements into design concepts, with each concept corresponding to several unique words. Further, this study extends the use of both LDA and topic visualization techniques for analyzing requirement topics. Creating topics allows designers to track engineering changes within most related requirements based on semantic correlations. Similarly, other adjacent topics containing closely related words may also be affected by any initial change.

Three research questions have addressed the issues of applying LDA to requirements management. For the first question, a detailed case study explores the feasibility of generalizing requirements into topics with different visualizations. Three industrial datasets are then implemented to demonstrate the performance of topic analysis for requirements. The second research question emphasizes the number of topics. Three techniques are utilized to estimate the necessary number of topics by comparing both perplexity and coherence scores. The initial finding reveals that topic merging occurs when fewer topics are assigned to the model. Fewer topics can provide a general visualization of the system, and more topics can provide lower-level design details. A trade-off decision is made based on purposeful interpretation and domain knowledge. In addition, the third research question focuses on the quality of each topic by implementing the LDAvis tool to visualize the topic-word distribution. The $\lambda$ value can be adjusted to improve the quality of each topic by selecting the most relevant terms. In sum, this study provides a framework to implement a supervised LDA model to capture the design information from requirements documents. The results indicate that while the generated topics are useful for some design information, they cannot stand alone in the design process apart from human intervention. To reach a desired performance, further study is required to explore different topic model structures.

Further research should explore different approaches or different variations of topic models to represent design documents in requirements management. By making additional assumptions, other methods may contribute to accurately capturing the semantics of unstructured requirements documents into subgroups. A comparison study could then be conducted to illustrate the performance differences with the baseline LDA model. Another option would be to label requirements into different categories; however, this would require training in a large and diverse repository, where the trained model can assess varied designs based on predetermined topics. Beyond enhancing the LDA model, the development of such a model could be used to inform design activities, such as conceptual design and geometric creation. The topic could be used to bridge domains that allow for both upstream and downstream analysis, a feat currently limited in design.

In line with the previous findings, a subsequent study compared the performance of different combinations of topic modeling and word embedding techniques to further improve the quality of each topic. We address two major challenges of analyzing requirement documents, including extracting information from short sentences and mapping topic-word correlations from domain-specific documents. Following an exploratory data analysis, the proposed framework combines topic modeling (e.g., LDA or GSDMM) with word embedding (e.g., sentence_BERT). We validate each model using either topic coherence or Silhouette scores. Our results indicate that both LDA BERT and GSDMM BERT achieve comparable results when compared to a single topic model. Although GSDMM is designed to cluster short texts, the results demonstrate that both LDA_BERT and GSDMM_BERT achieve similar results in generalizing design topics. We also show that both models can enhance the quality of each topic by including more relevant keywords. Overall, this study contributes to the goal of generating high quality design topics from requirements documents in building digital threads for smart manufacturing. In particular, the study demonstrates what types of analysis are critical to understanding complex system design topics.

Future work could apply this approach to diverse types of requirements documents. This proposed framework can be combined with the concept of ensemble learning. Hence, the final model would use a variety of topic models or word embedding algorithms to produce different vector representations to obtain more robust results. Furthermore, another aspect of this study could be combined with other techniques for automatically determining the number of topics.

**Point Cloud Classification**. In the second part of the study, the objective is to recognize the various categories of CAD models and subassemblies. This framework, however, focuses on the development of models for recognizing mechanical objects. The Meta-SeL algorithm combines both meta-learning and SAE to classify point clouds into ten predefined distinct model categories. Using the ShapeNetCore dataset, which simulates real manufacturing design data with additional part label information, this study shows that Meta-SeL can achieve a competitive level of accuracy to deep learning models.

To improve the model's generalization performance, several techniques have been developed for introducing noise into the training datasets. The combination of normalization and random noise (such as jittering or translation) provided more accurate prediction results. Moreover, for the purpose of understanding certain misclassification errors, we have visualized all the training SAE weighs into a subspace to show the similarities between each of the categories. Certain designs exhibit similar geometric resemblances, making it difficult for the proposed model to differentiate them. In comparison with state-of-the-art algorithms, our model achieves a high accuracy and utilizes only one epoch. In certain industries where time and computation power are critical resources, this could be particularly important. Comparing our model to deep learning applications, we find that we are more efficient at handling new data.

As the next step of this research, much work remains to ensure that this procedure can automatically predict mechanical sub-assemblies. As a further extension of our research, the future work will investigate how to improve the accuracy of classification by learning the representation function for each category.

**Linking Requirements to CAD Images**. The goal of this research is to develop a framework for automatically linking requirements and CAD - allowing engineers and designers to analyze how a change impacts one another. While much research exists on requirements-to-requirements and CAD-to-CAD analysis, minimal work exists on the linking of both. This is difficult as requirements (text) and CAD (geometric) operate in different domains. This research proposes a framework for linking said domains to bridge the gap between requirements and CAD.

We propose a framework for bridging gaps in and synthesizing multi-source data to facilitate knowledge acquisition and improve design efficiency. As image data may not always be available, we collected online images by using keywords filtered from requirements documents using POS tagging. To collect images from Google search results, a web scraping technique is used. Images are manually verified and modified according to the closest interpretation of requirements. The collected image dataset is verified and resampled to the same size. We demonstrate an improvement in model prediction by showing the top five most relevant requirements after fine-tuning the CLIP model. Testing images are selected from a variant design to assess the robustness of the model.

The major contributions of this work are threefold. First, we provide a method for constructing a synthetic image dataset representing the physical components of requirements. As image data is not always available, this technique enables a visual representation of requirements for tracking engineering change propagation. Secondly, using transfer learning, we combine prior knowledge with domain-specific information to understand the connection between requirements and images.

As a result of the learned correlations, similar mechanical components form out-of-distribution image datasets can be identified for identifying and interpreting requirements. Third, the predicted results illustrate the performance and limitations of the models by indicating the most relevant requirements for invariant designs. By taking photographs of different mechanical components and predicting the top requirements, engineers can determine which components are affected to minimize risks for a complex system.

Future work can be extended in several directions. Several CLIP model architectures and other industrial design documents should be considered. In CLIP model, various kinds of image and text encoders can be tested and compared. As simulation performance might differ based on the datasets, comparing various model architectures with publicly available design documentation may provide useful insights into distinct types of product designs.

# Bibliography

Ahmed, F., & Fuge, M. (2018). Creative exploration using topic-based bisociative networks. *Design Science*, *4*.

Alexiou, E., Upenik, E., & Ebrahimi, T. (2017). Towards subjective quality assessment of point cloud imaging in augmented reality, In *2017 ieee 19th international workshop on multimedia signal processing (mmsp)*. IEEE.

Andreou, A. S., Zographos, A. C., & Papadopoulos, G. A. (2003). A three-dimensional requirements elicitation and management decision-making scheme for the development of new software components., In *Iceis (3)*. Citeseer.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). Vqa: Visual question answering, In *Proceedings of the ieee international conference on computer vision*.

Arun, R., Suresh, V., Madhavan, C. V., & Murthy, M. N. (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations, In *Pacific-asia conference on knowledge discovery and data mining*. Springer.

Bach, F. R., & Jordan, M. I. (2006). Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, *7*(Oct), 1963–2001.

Bajaj, M., & Hedberg Jr, T. (2018). System lifecycle handler - spinning a digital thread for manufacturing, In *Incose international symposium*. Wiley Online Library.

Bajaj, M., Zwemer, D., Yntema, R., Phung, A., Kumar, A., Dwivedi, A., & Waikar, M. (2016). Mbse++ - foundations for extended model-based systems engineering across system lifecycle, In *Incose international symposium*. Wiley Online Library.

Ball, Z., & Lewis, K. (2019). Predicting multi-disciplinary design performance utilizing automated topic discovery, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Ball, Z., & Lewis, K. (2020). Predicting design performance utilizing automated topic discovery. *Journal of Mechanical Design*, *142*(12), 121703.

Bartels, R. H., & Stewart, G. W. (1972). Solution of the matrix equation ax+ xb= c [f4]. *Communications of the ACM*, *15*(9), 820–826.

Bello, S. A., Yu, S., Wang, C., Adam, J. M., & Li, J. (2020). Deep learning on 3d point clouds. *Remote Sensing*, *12*(11), 1729.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Bloebaum, C. L., & McGowan, A.-M. R. (2012). The design of large-scale complex engineered systems: Present challenges and future promise, In *12th aiaa atio conference and 14th aiaa/issmo ma&o conference, aiaa paper*.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering management*, *48*(3), 292–306.

Browning, T. R. (2015). Design structure matrix extensions and innovations: A survey and new opportunities. *IEEE Transactions on engineering management*, *63*(1), 27–52.

Castet, C. e. a. (2017). A point of view from mbse practitioners. *NASA JPL*.

Cellucci, T. A. (2008). Developing operational requirements. *US Department of Homeland Security*.

Cerpa, N., & Verner, J. M. (2009). Why did your project fail? *Communications of the ACM*, *52*(12), 130–134.

Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models, In *Neural information processing systems*. Citeseer.

Chen, C., Mullis, J., & Morkos, B. (2021). A topic modeling approach to study design requirements, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Chen, Y., Cheng, P., & Yin, J. (2010). Change propagation analysis of trustworthy requirements based on dependency relations, In *2010 2nd ieee international conference on information management and engineering*. IEEE.

Chen, Z. Y., Yao, S., Lin, J. Q., Zeng, Y., & Eberlein, A. (2007). Formalisation of product requirements: From natural language descriptions to formal specifications. *International Journal of Manufacturing Research*, *2*(3), 362–387.

Chen, Z. Y., & Zeng, Y. (2006). Classification of product requirements based on product environment. *Concurrent Engineering*, *14*(3), 219–230.

Cheng, Y., Zhang, Y., Ji, P., Xu, W., Zhou, Z., & Tao, F. (2018). Cyber-physical integration for moving digital factories forward towards smart manufacturing: A survey. *The International Journal of Advanced Manufacturing Technology*, *97*(1), 1209–1221.

Chung, F. R., & Graham, F. C. (1997). *Spectral graph theory*. American Mathematical Soc.

Clarkson, P. J., Simons, C., & Eckert, C. (2004). Predicting change propagation in complex design. *J. Mech. Des.*, *126*(5), 788–797.

Commons, W. (2020). File:latent dirichlet allocation.svg — wikimedia commons, the free media repository [[Online; accessed 16-July-2022]]. https://commons.wikimedia.org/w/index.php?title=File:Latent_Dirichlet_allocation.svg&oldid=469657330

Cross, N. (2021). *Engineering design methods: Strategies for product design*. John Wiley & Sons.

Danilovic, M., & Browning, T. R. (2007). Managing complex product development projects with design structure matrices and domain mapping matrices. *International journal of project management*, *25*(3), 300–314.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, *40*(2), 1–60.

David, M., & Rowe, F. (2016). What does plms (product lifecycle management systems) manage: Data or documents? complementarity and contingency for smes. *Computers in Industry*, *75*, 140–150.

Davis, J., Edgar, T., Porter, J., Bernaden, J., & Sarli, M. (2012). Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Computers & Chemical Engineering*, *47*, 145–156.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

de Paulo Faleiros, T., & de Andrade Lopes, A. (2016). On the equivalence between algorithms for non-negative matrix factorization and latent dirichlet allocation., In *Esann*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dor, L. E., Mass, Y., Halfon, A., Venezian, E., Shnayderman, I., Aharonov, R., & Slonim, N. (2018). Learning thematic similarity metric from article sections using triplet networks, In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Eckert, C., Clarkson, P. J., & Zanker, W. (2004). Change and customisation in complex engineering domains. *Research in engineering design*, *15*(1), 1–21.

Eng, N., Aurisicchio, M., & Bracewell, R. (2017). Mapping software augments engineering design thinking. *Journal of Mechanical Design*, *139*(5).

Eriksson, M., Morast, H., Börstler, J., & Borg, K. (2005). The pluss toolkit? extending telelogic doors and ibm-rational rose to support product line use case modeling, In *Proceedings of the 20th ieee/acm international conference on automated software engineering*.

Faghri, F., Fleet, D. J., Kiros, J. R., & Fidler, S. (2017). Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks, In *International conference on machine learning*. PMLR.

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse processes*, *25*(2-3), 285–307.

Frechette, S. P. (2011). Model based enterprise for manufacturing, In *Proceedings of the 44th cirp international conference on manufacturing systems*.

Fricker, S. (2010). Requirements value chains: Stakeholder management and requirements engineering in software ecosystems, In *International* *working conference on requirements engineering: Foundation for software quality*. Springer.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., & Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, *26*.

Fu, K., Cagan, J., Kotovsky, K., & Wood, K. (2013). Discovering structure in design databases through functional and surface based mapping. *Journal of mechanical Design*, *135*(3).

Giffin, M., de Weck, O., Bounova, G., Keller, R., Eckert, C., & Clarkson, P. J. (2009). Change propagation analysis in complex technical systems.

*Global model based enterprise market - industry analysis and forecast (2020-2027) - by deployment type, offering, industry and region.* (tech. rep.). (2020). https://www.maximizemarketresearch.com/market-report/global-model-based-enterprise-market/26999/

Gong, Y., Ke, Q., Isard, M., & Lazebnik, S. (2014). A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, *106*(2), 210–233.

Goyal, A., Law, H., Liu, B., Newell, A., & Deng, J. (2021). Revisiting point cloud shape classification with a simple and effective baseline. *International Conference on Machine Learning*.

Gregory, J., Berthoud, L., Tryfonas, T., Rossignol, A., & Faure, L. (2020). The long and winding road: Mbse adoption for functional avionics of spacecraft. *Journal of Systems and Software*, *160*, 110453.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, *101*(suppl 1), 5228–5235.

Groover, M., & Zimmers, E. (1983). *Cad/cam: Computer-aided design and manufacturing*. Pearson Education.

Guo, T., Xu, J., Sun, Y., Dong, Y., Davis, N., & Allison, J. T. (2018). Network analysis of design automation literature.*Journal of Mechanical Design*, *140*(10).

Gyory, J. T., Kotovsky, K., & Cagan, J. (2020). A topic modeling approach to study the impact of manager interventions on design team cognition, In *Asme 2020 international design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers Digital Collection.

Gyory, J. T., Kotovsky, K., & Cagan, J. (2021). The influence of process management: Uncovering the impact of real-time managerial interventions via a topic modeling approach. *Journal of Mechanical Design*, *143*(11).

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, *16*(12), 2639–2664.

Haskins, C., Forsberg, K., Krueger, M., Walden, D., & Hamelin, D. (2006). Systems engineering handbook, In *Incose*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition, In *Proceedings of the ieee conference on computer vision and pattern recognition*.

Hedberg, T., Lubell, J., Fischer, L., Maggiano, L., & Barnard Feeney, A. (2016). Testing the digital thread in support of model-based manufacturing and inspection. *Journal of Computing and Information Science in Engineering*, *16*(2).

Hedberg, T. D., Bajaj, M., & Camelio, J. A. (2020). Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads.*Journal of Computing and Information Science in Engineering*, *20*(1).

Hein, P. H. (2018). *Computational support for predicting requirement change volatility in complex system design* (Doctoral dissertation). Florida Institute of Technology.

Hein, P. H., Kames, E., Chen, C., & Morkos, B. (2021). Employing machine learning techniques to assess requirement change volatility. *Research in Engineering Design*, *32*(2), 245–269.

Hein, P. H., Menon, V., & Morkos, B. (2015). Exploring requirement change propagation through the physical and functional domain, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Hein, P. H., Voris, N., & Morkos, B. (2018). Predicting requirement change propagation through investigation of physical and functional domains. *Research in Engineering Design*, *29*(2), 309–328.

Helmer, R., Yassine, A., & Meier, C. (2010). Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design*, *21*(6), 647–675.

Hirshorn, S. R. (2017). Nasa systems engineering handbook revision 2. *Aeronautics Research Mission Directorate*.

Hofmann, T. (1999). Probabilistic latent semantic indexing, In *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval*.

Hofmann, T. (2013). Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*.

Htet Hein, P., Morkos, B., & Sen, C. (2017). Utilizing node interference method and complex network centrality metrics to explore requirement change propagation, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Huang, G., & Mak, K. (1998). Computer aids for engineering change control. *Journal of Materials Processing Technology*, *76*(1-3), 187–191.

Huang, Y., Chen, C.-H., & Khoo, L. P. (2012). Kansei clustering for emotional design using a combined design structure matrix. *International Journal of Industrial Ergonomics*, *42*(5), 416–427.

Hull, E., Jackson, K., & Dick, J. (2005). *Requirements engineering in the solution domain*. Springer.

Jacobi, C., Van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, *4*(1), 89–106.

Jawahar, G., Sagot, B., & Seddah, D. (2019). What does bert learn about the structure of language?, In *Acl 2019-57th annual meeting of the association for computational linguistics*.

Jiao, J., & Chen, C.-H. (2006). Customer requirement management in product development: A review of research issues. *Concurrent Engineering*, *14*(3), 173–185.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval.*Journal of documentation*.

Joshi, S., Morkos, B., & Summers, J. D. (2019). Mapping problem and requirements to final solution: A document analysis of capstone design projects. *International Journal of Mechanical Engineering Education*, *47*(4), 338–370.

Joulin, A., Maaten, L. v. d., Jabri, A., & Vasilache, N. (2016). Learning visual features from large weakly supervised data, In *European conference on computer vision*. Springer.

Joung, J., & Kim, H. M. (2021). Automated keyword filtering in latent dirichlet allocation for identifying product attributes from online reviews.*Journal of Mechanical Design*, *143*(8), 084501.

Jung, S., & Simpson, T. W. (2017). New modularity indices for modularity assessment and clustering of product architecture. *Journal of Engineering Design*, *28*(1), 1–22.

Kapurch, S. J. (2010). *Nasa systems engineering handbook*. Diane Publishing.

Karima, M., Sadhal, K., & McNeil, T. (1985). From paper drawings to computer-aided design. *IEEE computer graphics and applications*, (2), 27–39.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions, In *Proceedings of the ieee conference on computer vision and pattern recognition*.

Keller, R., Eckert, C. M., & Clarkson, P. J. (2009). Using an engineering change methodology to support conceptual design. *Journal of Engineering Design*, *20*(6), 571–587.

Keogh, E. J., & Mueen, A. (2010). Curse of dimensionality.

Kidono, K., Miyasaka, T., Watanabe, A., Naito, T., & Miura, J. (2011). Pedestrian recognition using high-definition lidar, In *2011 ieee intelligent vehicles symposium (iv)*. IEEE.

Kobayashi, A., & Maekawa, M. (2001). Need-based requirements change management, In *Proceedings. eighth annual ieee international conference and workshop on the engineering of computer-based systems-ecbs 2001*. IEEE.

Kodirov, E., Xiang, T., & Gong, S. (2017). Semantic autoencoder for zero-shot learning, In *Proceedings of the ieee conference on computer vision and pattern recognition*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kropsu-Vehkapera, H., Haapasalo, H., Harkonen, J., & Silvola, R. (2009). Product data management practices in high-tech companies. *Industrial Management & Data Systems*.

Kusiak, A., & Larson, N. (1995). Decomposition and representation methods in mechanical design.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259–284.

Lee, J., & Hong, Y. S. (2017). Bayesian network approach to change propagation analysis. *Research in Engineering Design*, *28*(4), 437–455.

Li, A., Jabri, A., Joulin, A., & Van Der Maaten, L. (2017). Learning visual n-grams from web data, In *Proceedings of the ieee international conference on computer vision*.

Li, G., Duan, N., Fang, Y., Gong, M., & Jiang, D. (2020). Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training, In *Proceedings of the aaai conference on artificial intelligence*.

Li, J., Tao, F., Cheng, Y., & Zhao, L. (2015). Big data in product lifecycle management. *The International Journal of Advanced Manufacturing Technology*, *81*(1), 667–684.

Lin, Y., Tan, Y. C., & Frank, R. (2019). Open sesame: Getting inside bert's linguistic knowledge. *arXiv preprint arXiv:1906.01698*.

Liu, N., & Stewart, W. J. (2010). Markov chains and spectral clustering, In *International workshop on performance evaluation of computer and communication systems*. Springer.

Liu, X., Wang, W., Guo, H., Barenji, A. V., Li, Z., & Huang, G. Q. (2020). Industrial blockchain based framework for product lifecycle management in industry 4.0. *Robotics and computer-integrated manufacturing*, *63*, 101897.

Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., & Pan, C. (2019). Densepoint: Learning densely contextual representation for efficient point cloud processing, In *Proceedings of the ieee/cvf international conference on computer vision*.

Liu, Y., Fan, B., Xiang, S., & Pan, C. (2019). Relation-shape convolutional neural network for point cloud analysis, In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*.

Lovmar, L., Ahlford, A., Jonsson, M., & Syvänen, A.-C. (2005). Silhouette scores for assessment of snp genotype clusters. *BMC genomics*, *6*(1), 1–6.

Lubell, J., Chen, K., Horst, J., Frechette, S., & Huang, P. (2012). Model based enterprise/technical data package summit report. *NIST Technical Note*.

Lyu, J., Akhavan Taheri Boroujeni, J., & Manoochehri, S. (2021). In-situ laser-based process monitoring and in-plane surface anomaly identification for additive manufacturing using point cloud and machine learning, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Madni, A. M., & Sievers, M. (2018). Model-based systems engineering: Motivation, current status, and research opportunities. *Systems Engineering*, *21*(3), 172–190.

Mao, S., Wang, B., Tang, Y., & Qian, F. (2019). Opportunities and challenges of artificial intelligence for green manufacturing in the process industry. *Engineering*, *5*(6), 995–1002.

Mavroeidis, D. (2011). Mind the eigen-gap, or how to accelerate semi-supervised spectral learning algorithms, In *Twenty-second international joint conference on artificial intelligence*.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

McLellan, J. M., Morkos, B., Mocko, G. G., & Summers, J. D. (2010). Requirement modeling systems for mechanical design: A systematic method for evaluating requirement management tools and languages, In *International design engineering technical conferences and computers and information in engineering conference*.

Mitchell, R. (2018). *Web scraping with python: Collecting more data from the modern web.* " O'Reilly Media, Inc."

Mocko, G. M., Summers, J. D., Fadel, G. M., Teegavarapu, S., Maier, J. R., Ezhilan, T., et al. (2007). A modelling scheme for capturing and analyzing multi-domain design information: A hair dryer design example, In *Ds 42: Proceedings of iced 2007, the 16th international conference on engineering design, paris, france, 28.-31.07. 2007*.

Model based enterprise report 2019 - global market outlook 2017-2026 - researchandmarkets.com. (2019). https://www.businesswire.com/news/home/20190328005548/en/Model-Based-Enterprise-Report-2019---Global

Mohammadi, F. G., Arabnia, H. R., & Amini, M. H. (2019). On parameter tuning in meta-learning for computer vision, In *2019 international conference on computational science and computational intelligence (csci)*. IEEE.

Mohammadi, F. G., Chen, C., Shenavarmasouleh, F., Amini, M. H., Morkos, B., & Arabnia, H. R. (2022). 3d-model shapenet core classification using meta-semantic learning. *arXiv preprint arXiv:2205.15869*.

Mori, Y., Takahashi, H., & Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words, In *First international workshop on multimedia intelligent storage and retrieval management*. Citeseer.

Mørkeberg Torry-Smith, J., Qamar, A., Achiche, S., Wikander, J., Henrik Mortensen, N., & During, C. (2013). Challenges in designing mechatronic systems.*Journal of Mechanical Design*, *135*(1).

Morkos, B., Joshi, S., & Summers, J. D. (2019). Investigating the impact of requirements elicitation and evolution on course performance in a pre-capstone design course. *Journal of Engineering Design*, *30*(4-5), 155–179.

Morkos, B., Joshi, S., Summers, J. D., & Mocko, G. G. (2010). Requirements and data content evaluation of industry in-house data management system, In *International design engineering technical conferences and computers and information in engineering conference*.

Morkos, B., Mathieson, J., & Summers, J. D. (2014). Comparative analysis of requirements change prediction models: Manual, linguistic, and neural network. *Research in Engineering Design*, *25*(2), 139–156.

Morkos, B., Shankar, P., & Summers, J. D. (2012). Predicting requirement change propagation, using higher order design structure matrices: An industry case study. *Journal of Engineering Design*, *23*(12), 905–926.

Morkos, B., & Summers, J. D. (2009). Elicitation and development of requirements through integrated methods, In *International design engineering technical conferences and computers and information in engineering conference*.

Morkos, B., & Summers, J. D. (2013). A study of designer familiarity with product and user during requirement elicitation. *International Journal of Computer Aided Engineering and Technology*, *5*(2/3).

Morkos, B. W. (2012). *Computational representation and reasoning support for requirements change management in complex system design* (Doctoral dissertation). Clemson University.

Nathan Hartman. (2018). *Today's benefits and challenges of a model-based enterprise*. Vertex. https://vertexvis.com/resources/blog/benefits-challenges-model-based-ent

Ncube, C., & Maiden, N. A. (1999). Pore: Procurement-oriented requirements engineering method for the component-based systems engineering development paradigm, In *International workshop on component-based software engineering*.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm, In *Advances in neural information processing systems*.

Nilsson, P., & Fagerström, B. (2006). Managing stakeholder requirements in a product modelling system. *Computers in Industry*, *57*(2), 167–177.

Nouacer, R., Djemal, M., Niar, S., Mouchard, G., Rapin, N., Gallois, J.-P., Fiani, P., Chastrette, F., Lapitre, A., Adriano, T., et al. (2016). Equitas: A tool-chain for functional safety and reliability improvement in automotive systems. *Microprocessors and Microsystems*, *47*, 252–261.

Pahl, G., & Beitz, W. (2013). *Engineering design: A systematic approach*. Springer Science & Business Media.

PMI. (2014). *Requirements Management a Core Competency for Project And Program Success* (In-Depth report). Project Management Institute.

Polanco, X., & San Juan, E. (2006). Text data network analysis using graph approach.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., & Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation, In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining*.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation, In *Proceedings of the ieee conference on computer vision and pattern recognition*.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, *30*.

Qiang, J., Qian, Z., Li, Y., Yuan, Y., & Wu, X. (2020). Short text topic modeling techniques, applications, and performance: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1427–1445.

Qiao, L., Efatmaneshnik, M., Ryan, M., & Shoval, S. (2017). Product modular analysis with design structure matrix using a hybrid approach based on mds and clustering. *Journal of Engineering Design*, *28*(6), 433–456.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision, In *International conference on machine learning*. PMLR.

Rajaraman, A., & Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Rangan, R. M., Rohde, S. M., Peak, R., Chadha, B., & Bliznakov, P. (2005). Streamlining product lifecycle processes: A survey of product lifecycle management implementations, directions, and challenges.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures, In *Proceedings of the eighth acm international conference on web search and data mining*.

Saaksvuori, A., & Immonen, A. (2008). *Product lifecycle management systems*. Springer.

Saidani, M., Kim, H., & Yannou, B. (2021). Can machine learning tools support the identification of sustainable design leads from product reviews? opportunities and challenges, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Salvador, S., & Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms, In *16th ieee international conference on tools with artificial intelligence*. IEEE.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schofield, A., Magnusson, M., Thompson, L., & Mimno, D. (2017). Understanding text pre-processing for latent dirichlet allocation, In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering, In *Proceedings of the ieee conference on computer vision and pattern recognition*.

Shankar, P., Morkos, B., & Summers, J. D. (2010). A hierarchical modeling scheme with non functional requirements, In *International design engineering technical conferences and computers and information in engineering conference*.

Shankar, P., Morkos, B., & Summers, J. D. (2012). Reasons for change propagation: A case study in an automotive oem. *Research in Engineering Design*, *23*(4), 291–303.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), 888–905.

Sievert, C., & Shirley, K. (2014). Ldavis: A method for visualizing and interpreting topics, In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*.

Socher, R., Ganjoo, M., Manning, C. D., & Ng, A. (2013). Zero-shot learning through cross-modal transfer. *Advances in neural information processing systems*, *26*.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, *29*.

Song, B., Meinzer, E., Agrawal, A., & McComb, C. (2020). Topic modeling and sentiment analysis of social media data to drive experiential redesign, In *International design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Srivastava, N., & Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. *Advances in neural information processing systems*, *25*.

Summers, J., & Morkos, B. (2013). Requirements evolution: Impact of functional and non-functional change on project success, In *International design engineering technical conferences and computers and information in engineering confernce*.

Summers, J. D., Joshi, S., & Morkos, B. (2014). Requirements evolution: Relating functional and non-functional requirement change on student project success, In *International design engineering technical* *conferences and computers and information in engineering conference*. American Society of Mechanical Engineers.

Tao, F., Zhang, L., Liu, Y., Cheng, Y., Wang, L., & Xu, X. (2015). Manufacturing service management in cloud manufacturing: Overview and future research directions. *Journal of Manufacturing Science and Engineering*, *137*(4).

Teh, Y. W., Newman, D., & Welling, M. (2007). *A collapsed variational bayesian inference algorithm for latent dirichlet allocation* (tech. rep.). CALIFORNIA UNIV IRVINE SCHOOL OF INFORMATION and COMPUTER SCIENCE.

Terwiesch, C., & Loch, C. H. (1999). Managing the process of engineering change orders: The case of the climate control system in automobile development. *Journal of Product Innovation Management: AN INTERNATIONAL PUBLICATION OF THE PRODUCT DEVELOPMENT & MANAGEMENT ASSOCIATION*, *16*(2), 160–172.

Tilstra, A. H., Seepersad, C. C., & Wood, K. L. (2012). A high-definition design structure matrix (hddsm) for the quantitative assessment of product architecture. *Journal of Engineering Design*, *23*(10-11), 767–789.

Ullman, D. G. (1992). *The mechanical design process* (Vol. 2). McGraw-Hill New York.

Ulrich, K. T. (2003). *Product design and development*. Tata McGraw-Hill Education.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T., & Yeung, S.-K. (2019). Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data, In *Proceedings of the ieee/cvf international conference on computer vision*.

Vafaeesefat, A., & ElMaraghy, H. A. (1999). *Data Reduction for Reverse Engineering of Free Form Surfaces*, 803–809. https://doi.org/10.1115/DETC99/CIE-9132

van Gemert, J. (2003). Retrieving images as text.

Veisz, D., Namouz, E. Z., Joshi, S., & Summers, J. D. (2012). Computer-aided design versus sketching: An exploratory case study. *AI EDAM*, *26*(3), 317–335.

Violante, M. G., Vezzetti, E., & Alemanni, M. (2017). An integrated approach to support the requirement management (rm) tool customization for a collaborative scenario. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *11*(2), 191–204.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*(4), 395–416.

Wallach, H. M., Mimno, D. M., & McCallum, A. (2009). Rethinking lda: Why priors matter, In *Advances in neural information processing systems*.

Wang, L., Liu, Z., Liu, A., & Tao, F. (2021). Artificial intelligence in product lifecycle management. *The International Journal of Advanced Manufacturing Technology*, *114*(3), 771–796.

Wang, L., Li, Y., Huang, J., & Lazebnik, S. (2018). Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(2), 394–407.

Wang, S., & Rohe, K. (n.d.). Don't mind the (eigen) gap.

Wang, Y., Zheng, L., Hu, Y., & Fan, W. (2018). Multi-source heterogeneous data collection and fusion for manufacturing workshop based on complex event processing, In *Proceedings of the 48th international conference on computers & industrial engineering (cie), auckland, new zealand*.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, *38*(5). https://doi.org/10.1145/3326362

Wardhani, R., & Xu, X. (2016). Model-based manufacturing based on step ap242, In *2016 12th ieee/asme international conference on mechatronic and embedded systems and applications (mesa)*. IEEE.

Wu, Z., & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *15*(11), 1101–1113.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes, In *Proceedings of the ieee conference on computer vision and pattern recognition*.

Xiao, H., & Stibor, T. (2010). Efficient collapsed gibbs sampling for latent dirichlet allocation, In *Proceedings of 2nd asian conference on machine learning*. JMLR Workshop and Conference Proceedings.

Xing, E. P., Jordan, M. I., Russell, S. J., & Ng, A. Y. (2003). Distance metric learning with application to clustering with side-information, In *Advances in neural information processing systems*.

Yang, Q., Yao, T., Lu, T., & Zhang, B. (2013). An overlapping-based design structure matrix for measuring interaction strength and clustering analysis in product development project. *IEEE Transactions on Engineering Management*, *61*(1), 159–170.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., & Guibas, L. (2016). A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, *35*(6), 1–12.

Yin, J., & Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering, In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining*.

Yu, T.-L., Yassine, A. A., & Goldberg, D. E. (2007). An information theoretic method for developing modular architectures using genetic algorithms. *Research in Engineering Design*, *18*(2), 91–109.

Zelnik-Manor, L., & Perona, P. (2005). Self-tuning spectral clustering, In *Advances in neural information processing systems*.

Zhou, F., Ayoub, J., Xu, Q., & Jessie Yang, X. (2020). A machine learning approach to customer needs analysis for product ecosystems. *Journal of Mechanical Design*, *142*(1).