Text Analytics to Inform Deviation Root Cause Analysis in Biomanufacturing

by

Lois E. Nersesian

B.S., Chemical Engineering
University of California Los Angeles, 2018

Submitted to the MIT Sloan School of Management and
Department of Chemical Engineering
in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Chemical Engineering

in conjunction with the Leaders for Global Operations program

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
MIT Sloan School of Management and
Department of Chemical Engineering
May 6, 2022

Certified by . . . .
Retsef Levi, Thesis Supervisor
Professor of Operations Research and Operations Management

Certified by . . . .
Richard D. Braatz, Thesis Supervisor
Edwin R. Gilliland Professor of Chemical Engineering

Accepted by . . . .
Patrick S. Doyle
Robert T. Haslam (1911) Professor of Chemical Engineering

Accepted by . . . .
Maura Herson
Assistant Dean, MBA Program, MIT Sloan School of Management

Text Analytics to Inform Deviation Root Cause Analysis in Biomanufacturing

by

Lois E. Nersesian

Submitted to the MIT Sloan School of Management and
Department of Chemical Engineering
on May 6, 2022, in partial fulfillment of the
requirements for the degrees of
Master of Business Administration
and
Master of Science in Chemical Engineering

Abstract

In biomanufacturing, product quality and safety are critical and there are many controls in place to ensure that processes are followed within the prescribed operating limits. However, deviations from these processes inevitably occur, sometimes requiring in-depth investigations to determine the cause and prevent recurrence. Understanding quality trends on the manufacturing line is also critical in preventing quality issues. At Amgen, a leading biotechnology company, results of such investigations are stored long-term but only in a partially structured manner, making it hard to leverage this historical data to enhance deviation investigation efficiency and study long term quality trends. The goal of this project is to use these historical records to draw insights into the investigation process and help increase the efficiency and accuracy of future deviation investigations and overall quality assurance. To achieve this, we use natural language processing tools to derive information from text describing deviations and causal factors. Several methods are explored, namely, unsupervised clustering using machine learning and natural language processing to identify and cluster similar causal factors, explicit text extraction which identifies known key terms such as equipment mentioned in the text, and process-dependent step classification which leverages reference documents describing the manufacturing process to assign records to process steps. The outputs of these methods are presented in a proof-of-concept tool which can be used to assist investigators. Our results indicate that all these methods have benefits and drawbacks but can be used together for maximal insights. Based on the status of each method, we suggest that Amgen work to create a tool to present potential causal factors to investigators immediately, incorporating clustering and text extraction methods after minor refinement, and continue to explore the potential of process-driven methodologies.

Thesis Supervisor: Retsef Levi
Title: Professor of Operations Research and Operations Management

Thesis Supervisor: Richard D. Braatz
Title: Edwin R. Gilliland Professor of Chemical Engineering

Acknowledgments

Thank you to everyone who has supported my work on this thesis and my journey of growth and learning over the past 2 years in the Leaders for Global Operations program.

I would like to thank Amgen for providing me with the opportunity to work on this project and being incredibly generous with providing access to data. To my supervisors, Mark DiMartino and Shea Watrin, thank you so much for providing complete support in exploring methodologies and guiding me to push the project further than I would have on my own. I would also like to acknowledge other Amgen coworkers who helped on the project including Sebastian Hanet and Kristina Bacon. I could not have done this work without your input, advice, and feedback. Finally, thanks to Cathy Champagne and all the LGO alumni at Amgen who helped organize the internship.

Special thanks to advisors Retsef Levi and Richard Braatz for guiding the project and providing advice and direction. It was invaluable in taking the project to the next level. To collaborators Josh Wilde and Evan Yao, the work presented here is very much a collaboration between us. Your creative ideas made this project compelling to work on. Thanks as well to the LGO program office for all the work that they do to coordinate internships and ensure a positive student experience throughout the program.

Finally, a big thanks to the friends and family who have supported me and believed in me more than I believed in myself. I cannot thank you enough for your love and support.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

This chapter describes the context of the project within Amgen's Quality organization. Next the project is introduced, outlining the problem statement, scope, goals, and approach. Then an overview of the project as a whole is provided, previewing the results and conclusions that are discussed in subsequent chapters.

1.1 Background

1.1.1 Amgen, Biomanufacturing, and Quality

Amgen is a biopharmaceutical and biotechnology company based in Thousand Oaks, California. Since their first FDA drug approval in 1989, they have been serving patients with their innovative medicines. Their therapeutic areas of focus cover cardiovascular disease, oncology, bone health, nephrology, and inflammation. Amgen is mainly focused on biologics but also has some small-molecule products.

This paper covers topics related to the manufacturing lines of biologics, also known as "biomanufacturing". These types of drugs are produced in living organisms such as mammalian cells or bacteria cells as opposed to small molecules which are chemically synthesized. Manufacturing biologics follows a similar, usually batched, process across products and companies (see Figure 1-1 for a simplified overview). First, before production begins, cells are genetically modified to carry the gene encoding the protein

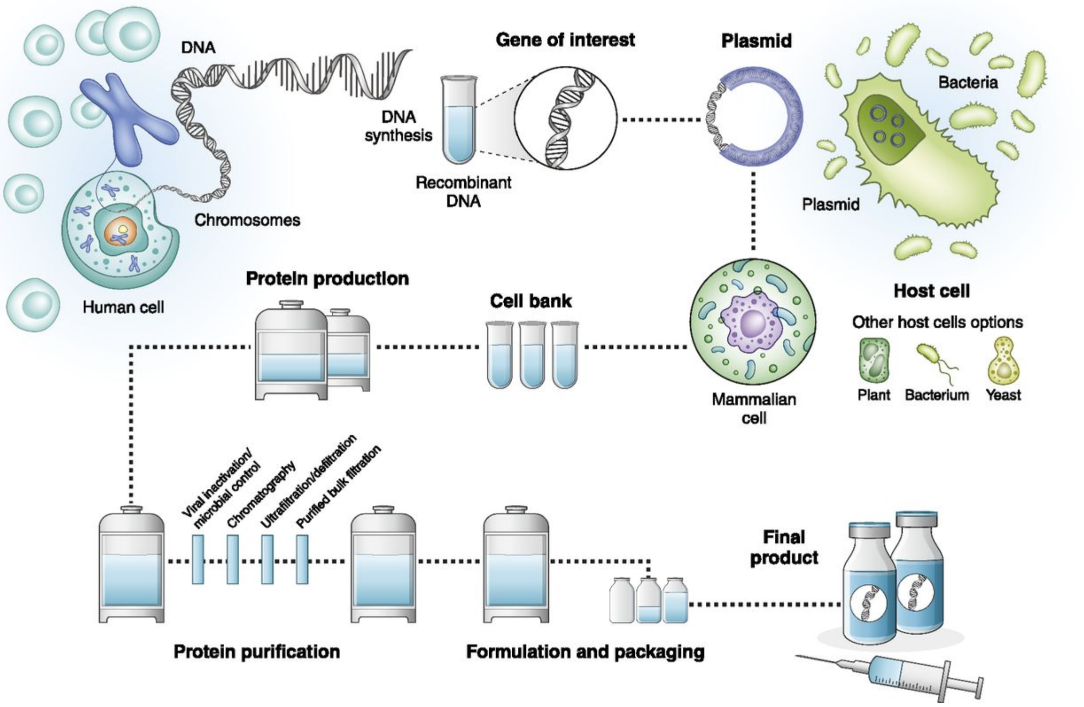


Figure 1-1: Overview of the manufacturing of biologics [28]

of interest, also called the active pharmaceutical ingredient (API). These cells establish a master cell bank which is used as a source throughout the life of the product. At the start of production, cells are taken from a working cell bank, which is seeded from the master cell bank, and go through several culturing steps to increase in number. Then, the cells may go through a series of bioreactors, continuing to grow in number, until reaching the main production bioreactor where the cells are induced to produce the protein of interest. Next is the harvest step where the cells are separated from the media containing the API. Finally, a variety of methods are used to purify and completely isolate the API, which may include ion exchange chromatography, hydrophobic interaction chromatography, and ultrafiltration/diafiltration among others. Once the API is isolated, manufacturing of the "drug substance" is complete and work on the "drug product" takes over. Drug product steps include formulation (for example, adding excipients), filling, and packaging.

At Amgen, and in any manufacturing setting, ensuring quality is of the upmost importance. The Quality organization within Amgen is independent of the manufacturing organization and serves to oversee the process and ensure high standards are met. In the pharmaceutical industry, this is particularly critical for regulatory compliance. The FDA and other organizations have standards and guidelines around biomanufacturing to ensure the biologics are safe, effective, and reliably produced. The FDA must approve the specific process parameters used in the manufacture of any drug as part of the overall approval for the drug. The Quality Management System ensures that procedures and practices are correctly followed for every manufacturing batch and allows for errors to be swiftly identified and dealt with. Technology systems serve a large role in the Quality Management System acting to catalog requirements, track issues, store documents, and perform other critical functions.

1.1.2 Deviation Investigation Process - Overview

As part of the Quality Management System and in compliance with FDA guidelines, it is critical to keep track of issues during the manufacturing process. At Amgen, these are called "deviations" and can be minor, causing little impact to product quality, or major, with the potential to cause serious issues including loss of the entire batch. In the case of major deviations, a team is assigned to investigate and perform a root cause analysis. In this process, they begin by brainstorming potential "causal factors" for the deviation. Next, they carefully investigate each causal factor to determine if it had a causal impact on the event. Once this is complete, they collectively determine the ultimate root cause of the deviation. The final step is to come up with corrective and preventative actions to stop this kind of deviation from happening again. The results of the investigation are detailed in a "deviation report" and stored in Amgen's computer systems as a permanent record. A more detailed description of the investigation process is provided in Chapter 3 of this thesis.

1.1.3 Deviation Trending Tool

Tracking deviations at Amgen is a very important process. Work has already been done to categorize deviations into groups, allowing investigators to discover and analyze trends. This helps fulfill regulatory requirements for companies to investigate such trends. Deviation groupings and trends at Amgen are displayed in a Spotfire dashboard accessible across the Quality organization but used mostly by deviation managers. The current implementation of this tool, which has been in place at Amgen since 2020, uses unsupervised clustering techniques to create the groupings. At a high level, this involves using natural language processing, specifically a deep learning language model, to encode the text describing the deviation and then clustering those texts together based on their themes. The language model considers the semantics of the text rather than the syntax or specific word usage. For Amgen's implementation, the model used is obtained pre-trained from the source and then fine-tuned on a corpus of deviation documents. The clusters generated are "nested" or "hierarchical" meaning there are layers of more refined clusters within larger clusters. Here, we will consider two important layers of clusters: 1) the "primary cluster" which is the biggest level of clustering with many deviations per group and 2) the "secondary cluster" which is the smallest level of clustering with only a few deviations per group. In the tool, given a new deviation, users can see what deviations are highly similar (in the same secondary cluster) and somewhat similar (in the same primary cluster). If there are many recent deviations in the same primary or secondary cluster, this indicates a potential trend that may need to be investigated. The current tool works well for Amgen's purposes but is not perfect. For example, deviations can only be assigned to one secondary cluster, but it is possible that a deviation may be related to more than one concept. Additionally, it uses a black-box language model which makes it difficult for general users to understand how the results are generated.

1.2 Project Definition

1.2.1 Problem Statement

Over the lifetime of manufacturing at Amgen, many deviation investigations have occurred. The reports generated from those investigations are stored indefinitely in

Amgen systems, but the information from them is not utilized to its fullest capacity. The deviation trend tool uses the basic description of the deviations but does not go further into other aspects of the reports. This information has huge potential to help improve the way deviations are handled and even prevent future deviations. The problem Amgen faces is how best to utilize the information gathered during deviation investigations. The hope is that it could serve to inform future investigations, making them more efficient and effective. In this thesis, we will investigate methods of gathering insights from this data, building on the work that has already been done to analyze deviations.

1.2.2 Scope and Goals

The scope of this project focuses on the root cause analysis portion of the deviation reports. Particularly, we will concentrate on the generation of potential causal factors. Currently, investigators will either manually search for related records and read through the reports to see what causal factors were proposed in the past or will propose causal factors completely independent of historical records. We hope that our solutions will, at minimum, streamline this process and ensure that investigators have easy access to relevant data. Determining the final root cause of a deviation is a very complex task that is difficult to predict. Since it is a less precise task, generating the list of causal factors has more potential to benefit from computer generated insights at this time. The outcome of any method will only assist the investigators. It is not intended to replace their efforts altogether.

There are two primary business outcome goals that the information from analyzing the causal factors for historical deviations hopes to achieve: 1) help investigators perform more accurate and speedy investigations and 2) provide overarching insights into the outcome of deviation investigations at Amgen, generally. Our goal is to generate proof-of-concept results towards these objectives that Amgen can later build upon.

1.2.3 Project Overview

The overarching approach to the project is to develop various methodologies that can add structure to the unstructured causal text for each deviation. To do this, we utilize free text descriptions of the deviations, causal factors, and root cause analyses as well as categorical variables available in our dataset. To demonstrate each strategy, we build a proof-of-concept tool in Tableau. The challenge in this project is both to determine the classification scheme that should be used to provide the most value as well as how to effectively fit the text into that classification scheme. Each method has a different classification scheme and/or strategy. The three methods that will be presented are:

1. Unsupervised Clustering: Use a deep learning language model to embed and then cluster causal factor text based on similarity. Common words appearing in each cluster serve as a cluster summary. Each causal factor is assigned to a single cluster.

2. Explicit Text Extraction: Gather a list of explicit names for key items such as equipment pieces and materials, and search for those items in the text. The results of the identification can be used to associate deviation items to causal items and create other meaningful connections. Each causal factor may contain many associated items.

3. Process-Dependent Step Classification: Associate root cause analysis and deviation text to specific steps in a linear manufacturing process using documentation describing each step as a source of step specific keywords. Each record is assigned to a singular step in the process.

In many ways, each method seeks to address the key issues that arise from the previous method. For example, the explicit text extraction method lacks hierarchy, so the process-dependent step classification focused entirely on how hierarchy and relationships between parts of the system can be incorporated. Additionally, the methods progress in terms of how much prior input goes into structuring their outputs.

For example, the clustering techniques are completely unsupervised and unspecified, text extraction is slightly more prescribed, and process-step classification is completely defined beforehand.

This thesis discusses each approach in detail, analyzing the quality of the results and the utility of the method to Amgen. In summary, we find that each method can provide value to Amgen but has some downsides. Unsupervised clustering is a promising method for grouping causal factors together and is relatively simple for Amgen to implement since it is similar to the methods used to cluster deviations. On the negative side, cluster summaries vary in precision and the cluster assignments are highly sensitive to model parameters. Text extraction methods mitigate these problems by standardizing the structure of the output, highlighting parts of the system that are known to be important, and allowing causal factors to be grouped in multiple dimensions. The item extraction also helps find interesting connections between parts of the manufacturing system, but it is limited by a lack of hierarchy. Our final method seeks to address this issue by targeting known hierarchies in the manufacturing system. However, the process-driven assignment techniques explored here are not successful in correctly matching records to steps in a process, though the idea of utilizing the known structure of the system still has promise. Additional research is needed to develop a robust technique, but the manual work required to define system processes may serve as a significant barrier to implementation.

In conclusion, we suggest that Amgen works on developing a tool or enhancing existing tools to show a list of causal factors for related deviations to investigators. Next, they can work to include results from unsupervised clustering and text extraction to provide additional details and insights. Process-driven techniques should continue to be explored at the proof-of-concept level before being incorporated into investigation tools.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Review

This chapter provides additional background discussing the broader context in which this project takes place. First, key tools used for natural language processing are covered with a focus on popular topic modelling tools and tools that will be used later in this thesis. Next, is a discussion of the challenges of ensuring quality within biomanufacturing and the advanced analytics tools that are being proposed to deal with those challenges.

2.1 Relevant Natural Language Processing Tools

Natural language processing (NLP) describes a set of computational techniques that allow for automatic analysis or representation of human language [5]. NLP tools have been implemented for such tasks as translating between languages, question-answering, and predictive text, among many others, and span both verbal and written language. For our purposes, the focus is on tools related to topic modelling which is a segment of NLP that looks to determine themes or topics from a set of text where the topics are generally unknown.

One of the most popular tools used in topic modelling is Latent Dirichlet Allocation (LDA). LDA is a statistical method that assigns documents to topics based on the words in the document. This is an example of "bag of words" approach, considering only the frequency of the words themselves and not the order of the words in the document. To use this method, you must specific the number of topics appearing in the set of documents. Then, using an iterative statistical approach, the model assigns each word in the corpus to a particular topic. The word assignments are then used to assign the documents based on the words they contain. While the number of topics must be specified, the theme of each topic is determined by the model [2]. LDA is commonly used across topic modelling applications. Some interesting examples include using LDA to topic model medical scientific publications then determine how AI tools are being developed across topics [27] and using LDA to topic model financial filings then comparing the results to stock market changes [8]. LDA is a useful tool but has some drawbacks. The number of topics may be unknown for the dataset thus requiring multiple runs of the algorithm to determine the optimal number. Additionally, LDA is non-hierarchical and cannot provide relationships between topics.

More advanced machine learning techniques have also been developed to represent human language in computer-readable forms. One of these commonly used models is Word2Vec which was developed by Google in 2013. Word2Vec turns individual words into vectors representing the semantic and syntactic meaning of the word. These vectors allow words to be "added" and "subtracted" with interesting results. For example, the vector for "king" minus the vector for "man" gives a vector equivalent to the vector for "queen" minus the vector for "woman" [22]. Word2Vec has been applied to a variety of NLP tasks. A couple of examples include using Word2Vec for sentiment classification of online shopping reviews [30] and determining whether news articles are related to a particular topic [12]. Word2Vec can be effective, but since it is a "bag of words" approach, the ordering of words is not considered. Additionally, Word2Vec cannot consider homonyms or words with multiple meanings. The output of Word2Vec is simply a vectorization of words. Additional tools must be subsequently applied for classification or clustering.

Other deep learning approaches go beyond Word2Vec by considering the context of the sentence in which the words appear. One such popular model is BERT (Bidirectional Encoder Representations from Transformers), also developed by Google and made available in 2018. What makes BERT superior is the fact that it considers the entire text at once rather than processing left-to-right or right-to-left as other models do. This allows BERT to understand the full context of the sentence. Two main training strategies were used to develop BERT. First is masked language modelling (MLM) where words in the text are masked and the model is tasked with predicting the word that appears in the masked position. Second is next sentence prediction (NSP) where the model is given pairs of sentences and asked to predict whether the second sentence is subsequent to the first in the original document. The training set is usually split with half positive pairs (from the same document) and half negative pairs (from random documents). The pre-trained BERT model can be used for transfer learning, that is, the pre-trained model can be downloaded and fine-tuned using a specific dataset while maintaining the gains from the original training. This saves time and computation and allows users to take advantage of pre-training with a large dataset [6]. However, the original BERT model has many millions of parameters and can be difficult to deploy in an industry setting. The solution is to use a compressed or distilled version of BERT called DistilBERT. This reduced model has half the number of parameters as BERT but retains 95% of the performance, making it effective and easier to implement [25]. BERT is a newer tool but has already been used in interesting contexts. For example, it has been used as a question-answering tool for digital invoices in Italy [1] and combined with a neural network to identify fake-news articles [14]. Like Word2Vec, BERT does not auto generate topics. It needs to be used with additional methods to organize the results into topics or generate other outputs.

LDA, Word2Vec, and BERT are all main tools used in NLP applications, but there are many other supporting tools and concepts that are relevant. First is the concept of "Term Frequency - Inverse Document Frequency" (TFIDF), which is a statistical tool for determining which words in a document are more important than others. Important words are those that distinguish the document from others, appearing commonly in the document of interest but rarely in others. Term frequency (TF) is calculated as the number of appearances of the word in the document divided by the total number of words in the document. Inverse document frequency (IDF) is the natural log of the number of documents in total divided by the number of documents containing the word. The final TFIDF score for a word in a document is its TF multiplied by its IDF. This equation makes it such that words appearing frequently in the document have higher scores and words appearing commonly across documents have lower scores [10]. The next concept to discuss is the process of lemmatization. Lemmatization of a word involves removing the inflectional endings allowing it to be grouped together with other words of the same underlying dictionary form or "lemma". For example, "ponies" is lemmatized to "pony", "organizing" to "organize", and "went" to "go". There are a variety of open-sourced packages available to complete this step. Improving the efficiency of lemmatization algorithms is an active area of research [10].

The final set of tools covered here are used for dimensionality reduction and clustering. These tools are not NLP specific but are commonly applied in topic modelling. The two steps together can take vector representations of words or sentences (such as the outputs of Word2Vec and BERT), reduce the number of dimensions of the vector, and group records together based on similarity. The tool used in this thesis for dimensionality reduction is UMAP (Uniform Manifold Approximation and Projection). Other popular options are tSNE and PCA (Principal Component Analysis). Dimension reduction is critical since clustering algorithms perform much better with fewer dimensions. Benefits of UMAP are that it preserves local and global structure and can go from very high dimensions to both medium and low dimensions. UMAP considers several parameters including number of nearest neighbors and minimum distance between nodes. One drawback is that UMAP has an element of randomness controlled by a random seed parameter [21]. The clustering tool used in this thesis is HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise). As the name suggests, this a density-based clustering algorithm which can form clusters of any shape. This is in contrast to popular methods such as K-means clustering which can only create round clusters based on a distance from a center point. HDBSCAN allows you to set a minimum cluster size and does not require the total number of clusters as a parameter [3].

2.2 Tools in Quality Assurance

In biomanufacturing, ensuring product quality is of the upmost importance, and in order to continuously improve on processes, root cause analysis of deviations is critical. However, root cause analysis investigations are particularly challenging to perform correctly in this space. One of the most common citations to pharmaceutical manufacturers from the FDA and other regulatory bodies is for inadequate root cause investigations [9]. Often, investigators will rush the process or stop investigating when they have reached the direct cause rather than going deeper into the real root cause. The direct cause is the action that directly resulted in the deviation, but the root cause is the actual issue that, if fixed, will prevent future occurrence. The best root cause analysis requires thorough investigation in order to be most effective [9].

In order to address the challenges with quality investigations, there are several concepts in place that, when implemented correctly, can help reduce the likelihood of deviations happening in the first place. These techniques can be applied across industries. One such tool is referred to as "Total Quality Maintenance" or "TQMain" which centers around frequent maintenance of not only equipment but all processes including environmental conditions, staff, methods, and materials. TQMain comes from integrating the equipment maintenance program with all other plant programs, forming a plant information technology system. This frequent attention can help avoid shutdowns of the line and ensure production schedules are met [23]. Another common quality assurance concept used in biomanufacturing is called "Quality by Design" or "QbD". This concept used in the development and manufacturing of pharmaceuticals builds quality into the process and product in a systematic and science-based manner. Under QbD, products are designed to ensure high quality when they are being manufactured. An example is to design the target protein with certain markers that make it simpler to purify and more robust to variances in fermentation conditions. An important part of QbD is the identification of critical quality attributes (CQAs) which define allowable ranges for a particular material or parameter that must be met to ensure product quality, safety, and efficacy. During manufacturing, it is preferred but not always possible to measure these CQAs to ensure the product will meet specifications. By considering quality from the start, you can reduce the likelihood and impact of costly deviations [29].

Regardless of the steps taken to reduce the occurrence of deviations, there will always be problems that occur. In many industrial manufacturing settings outside of biomanufacturing, work has been done to implement machine learning and other automated techniques to detect and predict defects. This is particularly useful when there are high tolerance requirements. The automotive industry is a big leader in this space. One example implementation used a learning process and pattern recognition strategy to classify outcomes as good or bad using data measured during automated manufacturing processes [7]. They tested their strategy on a process used in battery manufacturing as well as laser spot welding and were successful in identifying all "bad" outcomes [7]. Similar implements have been done in quality monitoring in plastic injection molding [26] and in the steel industry [17] pulling from both in-process monitoring data and materials data. Finally, another example from the automotive industry applies machine learning using data from in-process inspections to predict whether a certain car is likely to have certain defects detected during a later inspection [24]. In all of these examples, a large amount of in-process, continuous, and numerical data is used to generate predictions and detect defects. In biomanufacturing, processes are less precise and there is less mechanical automation making it more difficult to analyze these kinds of in-process data.

Advanced analytics solutions have also been explored in the area of investigations and root cause analysis. Research is currently being done to see how analytics can help target root cause investigations and the development of corrective and preventative actions (CAPAs). In one paper, researchers used multivariate data analysis (MVDA) to determine the root cause of issues during biomanufacturing scaleup [13]. They used data directly measured from the bioreactors including input measurements such as glucose levels and output measurements such as viable cell density, as well as time course measurements collected throughout the fermentation process. Their MVDA model is able to correctly identify the root cause of the issues, namely variances in input parameters. Other implementations apply machine learning techniques to quality investigations [11]. For example, sensors can be installed to monitor process parameters in real-time. Then, machine learning models can be used to predict CQAs in real time and determine whether a deviation has occurred. These models are trained using supervised techniques to learn how changes in parameters impact CQAs.

Both of these use cases [13] [11] focus on numerical measurements of particular parameters, but a variety of data can be used to inform root cause analyses. For example, one study in a non-pharmaceutical setting used a "big-data" approach to automatically predict root causes and suggest corrective actions. This study worked with multiple types of data from a variety of data sources and used it to develop a feature library describing quality problems. Then, they implemented machine learning algorithms to predict root causes [19]. Another approach used Bayesian networks to learn causal relationships in manufacturing processes. The model was developed using a simulated manufacturing line [18].

The applications of machine learning tools for quality control and assurance in biomanufacturing are just beginning to be explored but show significant potential. One idea is to work towards a "self investigating CAPA" where deviations are automatically captured and investigated then automated corrective actions are proposed. NLP tools could be used to monitor customer complaints, look for trends, and trigger an investigation. Other deep learning methods could be used to determine the impact of proposed CAPAs on the manufacturing process [20]. The use of advanced analytics techniques for quality in manufacturing, particular in biopharmaceuticals, is an active area of research with promising results to improve quality management.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Deviation Investigations - Process and Data

This chapter explains the deviation investigation process in depth and introduces the data that will be used in the rest of the thesis. How deviations are captured, investigated, and documented is explained in detail. Then, a discussion of the sources of all the data used is followed by a description of some key statistics regarding the main dataset that are used in Chapters 4 and 5. These descriptive statistics provide a perspective on how the data are distributed over some key axes.

3.1 Deviation Investigation Process

Whenever there is an issue in manufacturing, Amgen has processes in place to detect the problem, determine what went wrong, and make changes to prevent it from happening in the future. These issues are called "deviations" and are classified as either "minor" or "major" depending on their severity. Minor deviations are dealt with quickly while major deviations go through an intensive investigation process. Reporting, documenting, and investigating deviations is regulated and controlled by the FDA and other regulatory bodies. For major deviations, the process begins when a deviation is detected and entered into Amgen's deviation management system. The original observer fills out a form with basic information describing the deviation and a record is created. See Figure 3-1 for a summary of how deviations are captured and the deviation investigation process for major deviations.

After the initial intake process, a deviation manager is assigned. This person gathers a team of investigators including engineers, subject matter experts, functional area leads, process owners, and others, dependent on the situation. The team meets together and brainstorms ideas for what could have caused the deviation. These are called "causal factors" and are categorized into one of these "5M categories":

- Method: The causal factor relates to incorrect procedures
- Machinery: The causal factor relates to an equipment problem
- Materials: The causal factor relates to defective materials or material properties
- Human Performance: The causal factor relates to human action
- Environment: The causal factor is the environment where the issue took place

Organizing causal factors by 5M category helps ensure that the investigation team has considered all possible causes. Generally, teams try to propose at least one potential causal factor per category, where applicable.

After the list of causal factors is generated, the team investigates each one and determines whether it was a contributing factor to the deviation event. They may "confirm" the causal factor (it was definitely the cause), "disconfirm" the causal factor (it definitely was not the cause), or find it "probable" or "unconfirmed" (there is not enough evidence to prove or disprove that it contributed). Once the causal factors have been investigated, the team may use strategies such as "5-why's" to determine the final root cause of the deviation from the confirmed causal factors. The final investigative step is to develop corrective actions to ensure there are no lingering issues from the deviation and preventative actions to stop this kind of deviation from occurring in the future. These are known as Corrective and Preventative Actions (CAPAs).

The results of the investigation are written into a deviation report which is submitted to an Amgen system and permanently stored. The report will include a list

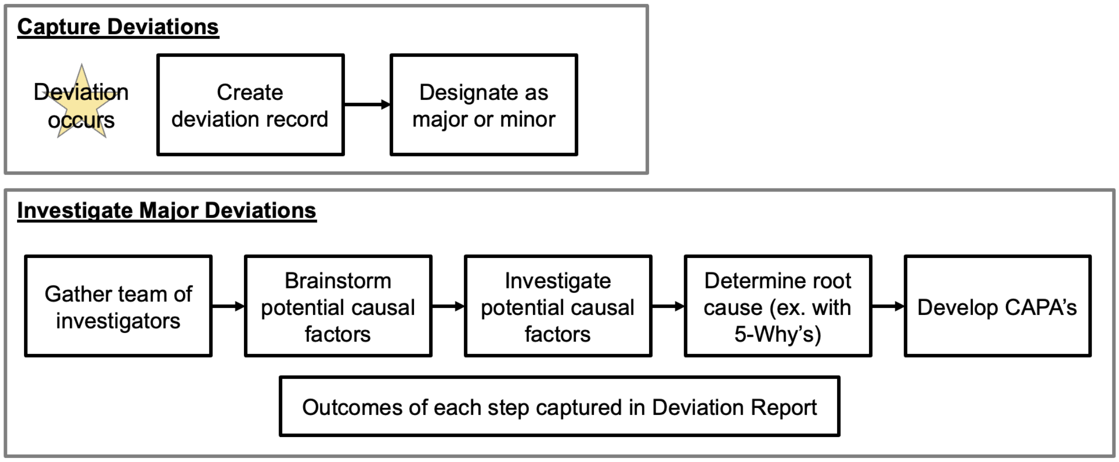

Capture Deviations
Deviation occurs
Create deviation record
Designate as major or minor
Investigate Major Deviations
Gather team of investigators
Brainstorm potential causal factors
Investigate potential causal factors
Determine root cause (ex. with 5-Why's)
Develop CAPA's
Outcomes of each step captured in Deviation Report

Figure 3-1: Overview of the deviation investigation process focusing on major deviation investigations.

of all potential causal factors that were investigated and a rationale for why they were confirmed, disconfirmed, or unconfirmed. There will also be a summary of the root cause analysis and a series of hierarchical codes describing the ultimate root cause. The most general code is the "root cause category" which puts the root cause into one of the 5M categories described above. Then there are two other levels of hierarchy, the "root cause code" and the "root cause subcode", which are more specific. The entire process from deviation to end of investigation takes about 20 to 25 days with a target to officially close the deviation within 30 days. See Appendix Figure A-1 for an outline of the contents of a standard deviation investigation report.

3.2 Data Sources

Amgen provided access to a suitable set of data sources including both historical reports for major deviations containing information about the deviation and the associated root cause analysis information as well as supplemental information related to their manufacturing system. The supplemental information particularly helps to decode language in the deviations such as acronyms and codes referring to documents or pieces of equipment. The methods described in Chapters 4 and 5 as well as the tool described in Chapter 7 draw from the same main dataset, while the method described in Chapter 6 uses a narrower set. All data used in the project are described below.

3.2.1 Core Datasets

The main source of data for the project is historical deviation reports for major deviations within Amgen manufacturing. For the purposes of this project, the scope of the data is narrowed to only include reports from drug substance manufacturing (the manufacturing of the active pharmaceutical ingredient) as opposed to drug product manufacturing (the rest of the activities involved in drug manufacturing). Additionally, only data from the last 10 years and from selected sites are included in the dataset. Any products falling into these filters are included. There is no filtering on product.

The reports were originally stored as PDF or Word files in Amgen's Quality Management System or in Amgen's company-wide document management system. Work was done by the Amgen team to download the files and extract the data from the causal factor tables included in all reports. Reports are generally written following a precise template which allowed the data to be repeatably extracted. See Appendix Figure A-1 for an overview of the template report sections. This report is finalized once the investigation is complete.

After extracting the information from the reports and combining with structured fields captured in Amgen's Quality Management System, the resulting dataset contains one row for each causal factor proposed in each major deviation within scope. This totals around 13,000 causal factors related to 2000 deviations. The dataset contains a mix of categorical structured variables as well as free-text unstructured data. Table 3.1 shows a summary of the fields available at the deviation and causal factor level. At the deviation level, categorical information is provided such as where the deviation took place referred to as the "area" (for example, "Purification") and the type of deviation (for example, "Process Parameter Excursion"), as well two free-text fields describing the deviation, one long and one short. The short description has on average around 13 words and the long description has around 245 words. Additionally, we are given information on the ultimate determined root cause category, root cause code, and root cause subcode.

	Deviation Level	**Causal Factor Level**
Unstructured Text	Short Description Long Description	CF Description Rationale
Categorical Variables	Area Site Deviation Type Root Cause Category Root Cause Code Root Cause Subcode	5M Category Confirmed?
Unique Identifiers	Deviation ID	Causal Factor ID

Table 3.1: Summary of fields available at the deviation and causal factor levels.

At the level of the causal factor, we are provided with a one sentence description of the causal factor (for example, "Standard operating procedure steps weren't followed properly"), a categorical field indicating whether or not the causal factor was confirmed as a root cause for the deviation, and a few sentence description of the rationale for why the causal factor was confirmed or ruled out. The description of the causal factor averages 13 words in length and the rationale averages 72 words. Finally, each causal factor is grouped into one of the "5M categories": machinery, materials, method, human performance, or environment. Table 3.1 summarizes these fields.

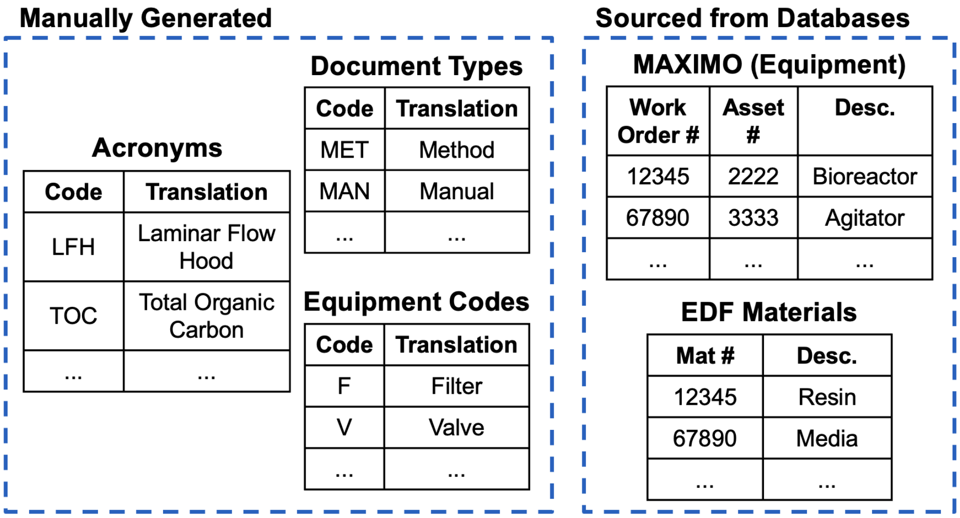
The final method, discussed in Chapter 6, uses a different dataset with a much narrower scope pulling only from Amgen's Quality Management System rather than from deviation reports. This dataset includes both major and minor deviations within one site, one product, and one unit operation, namely the production bioreactor, which is a key part of the drug substance manufacturing system. In this dataset, there is one entry for each deviation, and we only consider the overall root cause analysis field rather than specific causal factors. All the same fields describing the deviation as above are included. The newly utilized root cause analysis field is unstructured text. There are no other relevant new fields. This is a much smaller dataset containing around 30 times fewer records than the first dataset and about 1/5 of the number of deviations.

3.2.2 Additional Data Sources

A large variety of supplemental datasets are used to support this project. Firstly, we have access to Amgen's existing hierarchical deviation clustering, as described in Chapter 1, which can be used to group related deviations together. This is critical for understanding which deviations are similar to each other. The cluster assignments are updated on a regular cadence according to Amgen processes in order to incorporate new deviations.

Next, we have a variety of data sources that are used to transform codes in the free-text fields into more explicit text descriptions. These are generally fixed and independent of the core dataset. Later chapters will discuss exactly how these files are used to supplement and clean the free text fields described above. Figure 3-2 shows examples from all the files used for code translation. Some of these are manual files created from subject matter experts and manual inspection of the data. These contain translations for commonly used acronyms (for example, "TOC" is "total organic carbon"), document codes (for example, "MAN-01234" is a "manual"), and equipment codes (for example, "01-F-01234" is a "filter"). Access was also provided to various Amgen databases to gather additional information. One key system is MAXIMO, which Amgen utilizes as a database of work orders completed for pieces of equipment at their sites. It allows us to connect asset numbers and work order numbers referenced in the text to a description of the type of equipment (for example, "asset 01234" is a "temperature indicator"). Another system is the Enterprise Data Fabric within which we utilize a table matching material numbers to descriptions (for example, "material #9901234" is a "resin"). All of these files follow a similar structure with a column for the code and a column for the translation.

Finally, for the method described in Chapter 6, additional data sources are brought in to describe the production bioreactor process in detail and help understand how the steps fit together. This includes key SOPs (standard operating procedures) specific to the operation of the production bioreactor at the site. We also have access to diagrams of the processes and equipment that were verified with subject matter



Manually Generated

Acronyms

Code	Translation
LFH	Laminar Flow Hood
TOC	Total Organic Carbon
...	...

Document Types

Code	Translation
MET	Method
MAN	Manual
...	...

Equipment Codes

Code	Translation
F	Filter
V	Valve
...	...

Sourced from Databases

MAXIMO (Equipment)

Work Order #	Asset #	Desc.
12345	2222	Bioreactor
67890	3333	Agitator
...	...	...

EDF Materials

Mat #	Desc.
12345	Resin
67890	Media
...	...

Figure 3-2: Overview of the supporting files used for code translation with examples.

experts. All these documents were created when the process was first defined and have been continually refined and supplemented since. For our purposes, we consider these documents to be fixed and independent of the deviation process. There is no standard structure for these documents. They are free text usually organized into sections.

3.3 Descriptive Analysis of the Data

Now, we will look at the distribution of the causal factor and deviation dataset that will be used in Chapters 4, 5, and 7. This is used in combination with the deviation clusters dataset from the existing Amgen deviation trending tool. As a technical note, clustering of deviations was performed across the entire dataset of major deviations within Amgen over a period of time and included deviations not in our scope. Therefore, clusters may appear artificially small. To compensate for this, all secondary clusters containing only one deviation in our dataset are removed from the analysis. In the future, when the full dataset is used, there will never be a secondary cluster with only one deviation as the clustering algorithm is set to a minimum cluster size of two.

In our dataset, there are around 90 primary clusters with an average of 21.5 deviations per cluster and about 400 secondary clusters with an average of 3.3 deviations

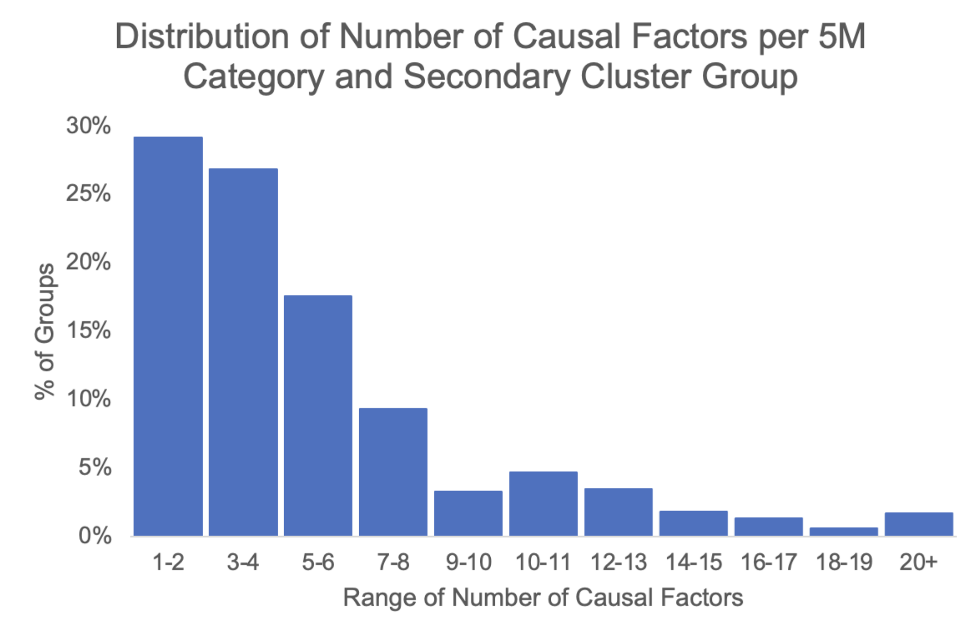


Figure 3-3: Histogram of the number of causal factors within a secondary cluster and 5M category group. The x-axis is the range of number of causal factors and the y-axis is the percentage of groups falling in that range.

per cluster. The maximum number of deviations in a secondary cluster is 35 but 99% of secondary clusters have less than 10 and 90% have less than 6.

Each deviation has 7 causal factors on average which is consistent with the business process where investigators try to propose at least one per 5M category. At the secondary cluster level, there is an average of 23.1 causal factors. Separated by 5M category the average is 5.4 causal factors per 5M category within a secondary cluster. This is a critical metric since users of a potential tool (described in detail in Chapter 7) will look at information for one secondary cluster at a time and likely will choose to read the causal factors by 5M category. Figure 3-3 shows the distribution of the number of causal factors within a group (secondary cluster and 5M category). Outliers above 20 are displayed as one group. The maximum value for any one group is 91 records, but this is a far outside the regular range. 98% of groups have less than 20 causal factors, 86% have less than 10 and, 56% have less than 5.

Across all proposed causal factors, 23% are confirmed as the root cause of the deviation. For the purposes of this analysis, "confirmed" is defined has having a status

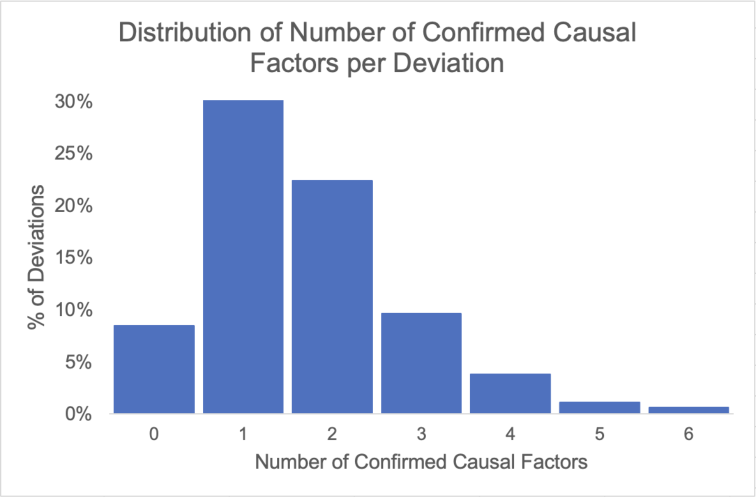


Figure 3-4: Histogram of number of confirmed causal factors per deviation.

of "confirmed", "probable", or "unconfirmed" per alignment with Amgen subject matter experts. Nearly all deviations have at least one confirmed causal factor (92%) though a handful have none. On average there are 1.7 confirmed causal factors per deviation. Figure 3-4 shows the distribution of number of confirmed causal factors per deviation. In some outliers, there have been as many 6, but the vast majority have 1 or 2.

Finally, we will look at root cause codes. As a reminder, root cause codes are broken into three parts: the root cause (RC) category which mirrors the 5M category, the RC code which is more specific, and the RC sub-code which is even more specific. We found that across the 5 possible categories there are 21 distinct codes and 54 distinct subcodes that are used. Table 3.2 shows how causal factors are distributed within 5M categories and how deviations are distributed within root cause categories. We can see that Method and Human Performance based causal factors and root causes are more likely to be proposed and chosen as the root cause. We can also see that the two distributions are similar, indicating that causal factors tend to be equally likely to be confirmed as the root cause though it skews slightly towards Method root causes.

5M Category	**% of Causal Factors**	**% of Deviation Root Cause Categories**
Method	33%	41%
Human Performance	28%	22%
Machinery	20%	17%
Material	12%	9%
Environment	8%	8%
None	n/a	2%

Table 3.2: Distribution of causal factors within 5M categories compared to the distribution of root cause categories for deviations

Chapter 4

Unsupervised Clustering

This chapter introduces our first methodology to transform and understand the text data. This involves the use of machine learning language models and clustering algorithms to group together causal factor descriptions based on their textual meanings. Achieving this requires a number of steps to first prepare the data, then apply the clustering algorithm, and then interpret the results. The application of the clustering algorithm involves searching and choosing optimal hyperparameters. Our discussion of the results focuses on how changing these parameters impacts the quality of the output.

4.1 Methods

The overall approach uses unsupervised clustering techniques to group causal factors together based on their unstructured text parts. Figure 4-1 shows a high-level overview of the steps and assets involved. The final output of a single clustering run is an assignment of each causal factor to a causal factor cluster, a 2-dimensional mapping of each causal factor, and a few words summary describing each cluster. We will commonly refer to "clustering runs" which are defined as an execution of the clustering algorithm with a specific set of hyperparameters, as well "clusters" which are a group of causal factors generated within a "clustering run". Below, we will discuss how clusters are generated as well as some key methods that will be used to assess the

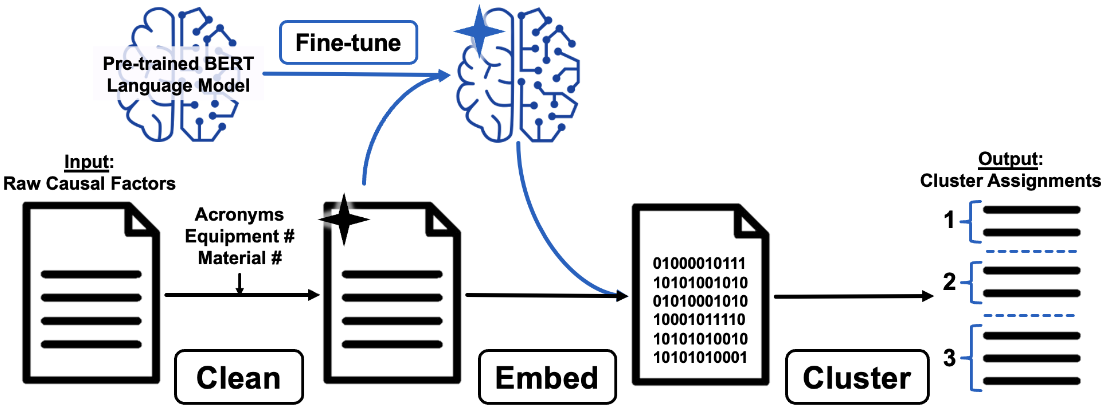


Figure 4-1: Overview of the unsupervised clustering process.

quality of the clustering runs. These assessment methods are used in the results section to compare clustering runs using different hyperparameters.

4.1.1 Data Cleaning

The first step of the clustering process is to clean the data by removing characters that the model will not process (such as special characters) and otherwise pre-process to optimize the final output. Using source files described in Chapter 3, we conduct three steps:

1. Remove acronyms: Every acronym appearing in our dictionary of acronyms is identified in all text fields and replaced with its literal interpretation. For example, "SIP" is replaced with "steam in place". This helps standardize the language used across text and makes it more interpretable for the model.

2. Mask product names: Specific product names are not relevant to clustering as we want causal factors to be grouped agnostic of product. Therefore any occurrence of a product name, such as "Enbrel", is replaced with "product".

3. Remove codes: Causal factors often contain references to equipment, documents, and other assets using codes rather than plain language. While a subject matter expert will understand what the text is referring to, the language model needs to have these codes translated. In order to achieve this, we identified categories

of numeric codes existing in the dataset and wrote regular expressions to isolate these codes in the text. Next, each category was assigned an action. The code may be removed from the text entirely if it is not relevant to the meaning of the text (for example, a specific batch number), or may be replaced with its category name when the code is important but not specific (for example, "Rm 1234" can be replaced with "room"), or may be translated when the particular item that the code is mentioning is important (for example, "EQ #12345" can be translated to "flask"). Translations are generally sourced from Amgen's MAXIMO database of equipment numbers and their Materials table of material numbers. These are discussed in more detail in Chapter 3.

To prepare for future steps, lemmatized versions of all text fields are created. In the lemmatization process, all stop words are removed and inflected forms of words are simplified. For example, the lemmatized version of "Problems occurred on the line" would be "problem occur line".

A final processing step used to help the model embedding process is the isolation of causal phrases in the causal factor text. Many causal factors are phrased in such a way to include reference to the deviation for which they were proposed. For example, "incorrect installation of the valve caused the leak in the bioreactor". This kind of wording may cause the language model to group causal factors together based on the deviation-related text rather than the actual cause. In this example, we want the text to be grouped based on "incorrect installation of the value" rather than "leak in the bioreactor". Therefore, we process the causal factor text to remove causation words and the text following or preceding them depending on the causation word. For example, "X caused Y" would reduce to "X" and "Z was caused by W" would reduce to "W".

4.1.2 Model Training

After the data are cleaned, the data are used to fine tune the pre-trained language model, in this case, DistilBERT. Fine tuning follows a similar strategy to the "Masked

Language Modeling" and "Next Sentence Prediction" training that was used to train BERT initially [6]. It is important to note that we are utilizing transfer learning, and the techniques here do not replace the existing training on the model. Instead, they merely fine-tune the existing model, making it more specific to this implementation. We are not training the model from scratch.

The general principle of our fine-tuning methodology is to give the model a set of positive examples, namely pairs of similar texts, and a set of negative examples, namely pairs of dissimilar texts. Similar texts should be embedded such that they have high cosine similarity and dissimilar texts should have low cosine similarity. Loss is calculated based on the difference between the current calculated cosine similarity and the goal, either 1 for similar texts or 0.01 for dissimilar texts. Figure 4-2 shows an example of positive and negative pairs being generated for a set of causal factors. One set of positive and negative examples is created for every causal factor document, so below, generating positive and negative examples is discussed from the perspective of a single causal factor. For our dataset, the positive pairs are created based on the assumption that a causal factor description text and its corresponding rationale contain similar ideas. The positive pairs are:

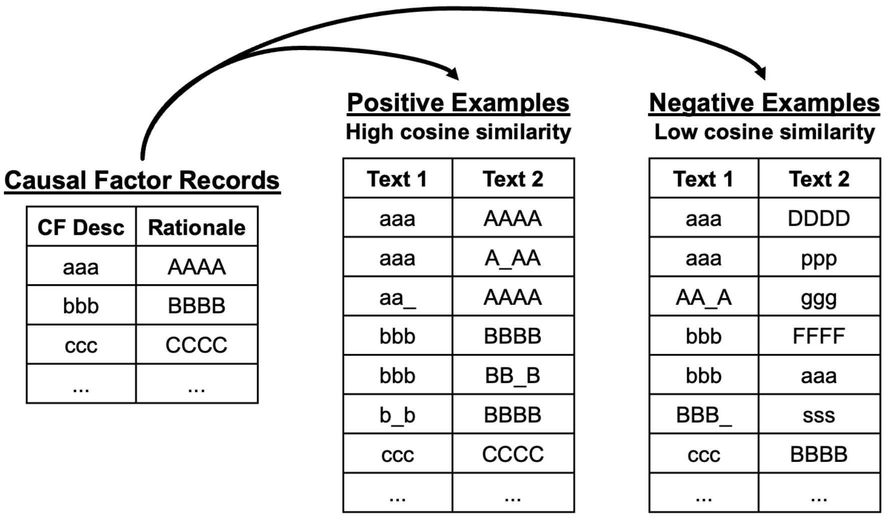
1. Causal factor description and corresponding rationale

2. Causal factor description and corresponding rationale masked

3. Causal factor description masked and corresponding rationale

Masked fields (like "rationale masked") are the same as their unmasked fields but with 15% of the words replaced with the "[MASK]" tag. This means that the text "Manufacturing associate did not follow directions in the procedure" might become "Manufacturing [MASK] did not follow directions [MASK] the procedure". This helps reduce overfitting.

The negative examples pair the causal factor or rationale text with a causal factor or rationale text randomly chosen from the dataset rather than with a text from its own document. This is based on the idea that two random texts selected from the



Causal Factor Records

CF Desc	Rationale
aaa	AAAA
bbb	BBBB
ccc	CCCC
...	...

Positive Examples
High cosine similarity

Text 1	Text 2
aaa	AAAA
aaa	A_AA
aa_	AAAA
bbb	BBBB
bbb	BB_B
b_b	BBBB
ccc	CCCC
...	...

Negative Examples
Low cosine similarity

Text 1	Text 2
aaa	DDDD
aaa	ppp
AA_A	ggg
bbb	FFFF
bbb	aaa
BBB_	sss
ccc	BBBB
...	...

Figure 4-2: Example of the generation of positive and negative examples used in DistilBERT model training. Underscores in text indicate masking.

dataset will likely be "far" or very different from each other. Examples of the negative pairs include:

1. Causal factor description with a random rationale
2. Causal factor description with a random causal factor description
3. Rationale masked with a random causal factor description

To increase the likelihood that the randomly selected texts are distinct in topic from the current text, examples are selected from a list of other causal factor documents with different 5M categories and from different deviations. In total, the positive and negative pairs provide 10 examples per causal factor with which to train the model. When training, only a few epochs are used to prevent overfitting.

The methods described can be used to fine-tune the base DistilBERT model available freely online, but they can also be applied to Amgen's existing model which started from the DistilBERT model and then fine-tuned on deviation data. This model, which we will call the "deviation-trained model", is currently used by Amgen to cluster deviation text and was trained in a similar fashion to what is described above but using deviation text rather than causal factor text. If we think about the model as having layers of training, the DistilBERT model has one layer, the base layer, and the deviation-trained model has two layers, the base layer and the deviation-trained layer. Our training would add another layer on top of those. Switching between the DistilBERT and deviation-trained models is the equivalent of swapping the "Pre-trained BERT Language Model" in the pipeline detailed in Figure 4-1 with the deviation-trained model. The reason that we might want to do this is that the deviation-trained model is already familiar with Amgen vocabulary. Additionally, it was trained with a much larger dataset than we have access to. In analyzing the results of unsupervised clustering, several models will be compared against each other. The models differ in the "layers" of training that they have been through:

1. Original DistilBERT model (1 layer: base training)
2. Existing deviation-trained model (2 layers: base training and deviation training)
3. Model starting from DistilBERT and trained on causal factors (2 layers: base training and causal factor training)
4. Model starting from deviation-trained model and trained on causal factors (3 layers: base training, deviation training, and causal factor training)

4.1.3 Embedding and Clustering

After the causal factors have been cleaned and the language model has been trained, the model is used to embed the clean causal factor text into 768-dimensional vector space. There are a variety of options for text to embed including the causal factor text alone, the causal factor text with the rationale, fully cleaned text, and less processed text. The results of using each in the pipeline are compared in the Results section.

To turn the embeddings into clusters, we first use the UMAP package to greatly reduce the dimensionality of the embeddings. UMAP takes the following input parameters: number of neighbors, number of components to reduce to, random state, and minimum distance [21]. After the dimension reduction, we use the HDBSCAN package to cluster the causal factors together. HDBSCAN takes a value for the minimum cluster size as the only parameter [3]. The output of these steps is a cluster assignment for each causal factor. Some will be categorized as "noise" if they do not fit well into any cluster. Text categorized as noise is considered "unclustered" and is not necessarily similar to other text categorized as noise. This clustering technique can also be performed on a subset of causal factors, for example, within 5M category. When clustering within a subset, the dataset is first partitioned, then the clustering is performed on each set entirely independently, and finally the results are combined.

One downside to using UMAP as the dimension reduction tool is the fact that it uses a random seed. Changing the random seed can cause the cluster assignments to vary greatly. Therefore, we developed a simple algorithm to combine cluster assignments from multiple random runs where all other parameters are identical. This method takes some causal factors classified as noise and adds them to a cluster and other causal factors that are clustered in different ways and sets them as noise. See Figure 4-3 for a flowchart describing the reassignment process for each causal factor. The process starts by taking one run of the clustering process (A) and mapping it to a second run (B). The mapping will show for each cluster in B which cluster in A it is most associated with. A cluster in B (B1) is mapped to a cluster in A (A1) if a plurality of the causal factors in B1 appear in A1. Once the mapping is established, the assignment can occur. For each causal factor, if it is noise in A and not B, the map is used to assign it to the A cluster mapped with its B cluster. Alternatively, if the A value does not match the B value's A mapping, this indicates a lack of confidence and the causal factor is assigned as noise. A cluster in B only maps to a cluster in A if the relationship is strong enough as defined by several parameters. This entire process can be performed recurrently in order to combine more than two clustering runs. For example, after B is mapped onto A, C can be mapped onto the combination of B and A.

Following the identification of clusters, steps are taken to determine a set of summary words to describe each cluster. This will be helpful for an investigator to understand what the cluster is about and will also be used to help determine the quality of the clustering. To achieve this, we first identify the top 10 words

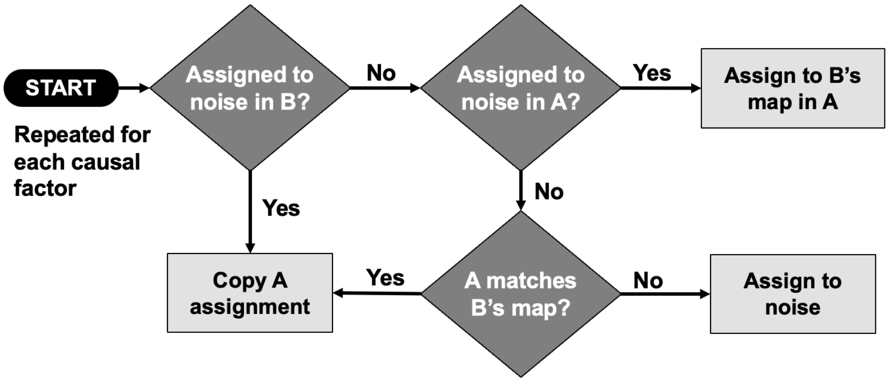


Figure 4-3: Flowchart describing the process of assigning a causal factor to a cluster by combining information from two cluster runs denoted A and B. This assumes a mapping exists from B clusters to A clusters.

associated with each cluster over other clusters using the TFIDF (term frequency inverse document frequency) score for each word [10]. According to TFIDF scoring, a word gets a high score if it appears commonly in the causal factors within the cluster, but it decreases if that word also appears commonly in all causal factors. The method is performed on the lemmatized version of the text so it excludes stop words and groups together inflected forms. Once we have the list of top 10 words, we count how many causal factors in the cluster contain each word. Words appearing in less than 10% of causal factors in the cluster are removed and the word are reordered based on their appearance percentage. The result is 1 to 10 words in order of importance, which can be used to summarize the topic of each cluster.

4.1.4 Clustering Evaluation Methodology

Since our data are unlabeled, we cannot directly measure the accuracy of the clustering method. Instead, we use a set of quantitative and qualitative methods to assess overall quality. The following sections describe the main methods used. We can also manually review the different clusters to check for quality but this will not be presented as a key method to distinguish between clustering run results. (As an important note, the methods described here are only applicable in the context of the particular unsupervised clustering methods presented above and will not be used in future chapters to evaluate their results.)

Basic Statistics

Some simple descriptions of the clusters, namely percent noise and number of clusters, are used to understand the clusters at a basic level. Generally, we prefer clustering runs that produce a lower percentage of causal factors assigned as noise. For number of clusters, the exact number is generally not relevant. However, in practice, a very small number of clusters indicates very poor clustering.

2-D Projection

UMAP is used to reduce the dimensionality of the causal factor embeddings into two dimensions so they may be graphed in 2-D. This projection is useful in qualitatively visualizing how effective the language model is at separating and grouping the causal factors. A poor model will embed all causal factors similarly into what looks like a single cloud whereas a better model will have clearer separation.

We can also use the 2-D projection to see how keywords are distributed within the clusters. From the 2-D mapping of causal factors, we plot the average center point of each cluster. Next, we filter to causal factors that contain a particular keyword, for example "training". Each cluster's center point is graphed as a dot, where the size of the dot depends on the number of causal factors within it that contain the keyword. High-quality clusters will map causal factors with the keyword close to each other and concentrate them in a few clusters. Results can be qualitatively compared to the distribution of generic words like "incorrect" which we would expect to be distributed across many clusters.

5M Entropy and Cluster Diversity

Quantitative measures can also be used to compare clustering runs. The first that we discuss is "5M entropy" which compares the distribution of 5M categories in the overall dataset with the distribution in each cluster. Entropy, in this context, is the

Kullback-Leibler divergence or "relative entropy" and is defined by:

$$D_{\mathrm{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right)$$

where D_{KL} is the relative entropy, P is the reference distribution (distribution of 5M categories in the general dataset), Q is the distribution of 5M categories in a particular cluster, and $\mathcal{X}$ is the set of all 5M categories [15]. Ideally, we want clusters with high entropy because this indicates that they have a high concentration of one or a couple of 5M categories and we know that causal factors in the same 5M category are more similar than those in different 5M categories. To assess the clustering run overall, we can take the average entropy over all clusters.

Our second quantitative measure is "cluster diversity" which is calculated at a deviation level. We define this metric as the number of non-noise unique clusters among a deviation's causal factors divided by its total number of non-noise causal factors. A value close to 1, the maximum, indicates high diversity in causal factors within the deviation. Overall, we want deviations within the clustering run to have high diversity. Low diversity could indicate that the model has clustered based on words related to the deviation rather than the cause.

Summary Word Frequency

The final metric of assessing cluster quality uses cluster summary words identified for each cluster, looking at how often they appear in causal factors in the cluster. Ideally, a cluster should have only a few key words that appear very commonly in its causal factors. We measure this using two features: how often the most common word appears and the difference between how often the top word appears and how often the fifth highest word appears. For each feature, we rank the clusters within the cluster run and then graph the cluster's rank against its feature value. In a graph showing multiple clustering runs, better runs will be above and to the right of poorer cluster runs as this indicates their clusters have higher values. Additionally, we can count the number of generic words appearing in cluster summaries for each run. Generic words are words such as "staff", "environment", or "equipment". Better clustering runs should have fewer generic words.

4.2 Results

Using the methods described in the previous section, the data are cleaned and various clustering runs performed using a mix of parameters. The following sections analyze the impact of changing parameters by comparing the results using the methods described in the "Clustering Evaluation Methodology" section. After providing an overview of the clustering results, the use of different language models are compared, both fine-tuned and un-fine-tuned. Next, we look at the impact of different text cleaning methods on the final output. Finally, we compare the results of clustering at different levels of detail and the impact of combining clustering runs using different random seeds.

4.2.1 Overview of Clustering Output

To set the stage, we first discuss some general findings about the clustering output by itself, taking one clustering run as representative of the output of clustering runs generally. In this run, we generated a total of 89 clusters. The median number of causal factors per cluster is 64 with a very wide range from around 30 to around 700 causal factors in some clusters.

Looking at the summary words generated for these clusters, we see mixed results. In many, if not most, cases the set of summary words are clear and interpretable (for example, "filter, cartridge, defective" or "pipette, calibration".) In others, the interpretation is less clear or is too vague to be especially useful (for example, "change, product" or "equipment, malfunction"). In particular, the very large clusters tend to generate poor cluster summaries (for example, "column" or "laboratory"). Manual inspection of these clusters indicates that they do group together causal factors of similar topics, but our summary word extraction method was not able pick out the important distinguishing keywords. Another phenomenon that we see in the results is that the cluster summaries can be at different levels of specificity. For example, one cluster summary simply says "pH, probe" while another includes "pH, probe" and several other words to describe what went wrong with the pH probe. It is possible that these clusters should be been combined or that the more generic cluster should have been split into smaller, more specific clusters.

When we look specifically at how some causal factors were clustered, we also find some interesting results. Firstly, there are clear examples showing where the causation word filtering step performed during the text cleaning process is important. For example, there is a cluster focused on issues with agitators, but the causal factor "Agitator direction changed due to motor malfunction" is not clustered in it. This is because our causation word filtering reduced this to simply "motor malfunction" and it was instead grouped into a cluster about equipment failure. This is a success since the cause being captured here is not with the agitator. The discussion of the agitator is merely a restatement of the deviation. However, sometimes groupings cannot be justified, particularly when looking at some records that are categorized as noise. For example, there is a cluster about analyst training that contains causal factors such as "Analyst not trained" and "Analyst involved in sample testing was not trained", but the very similar causal factor "Analyst involved in sample collection was not trained" is classified as noise. There is no explanation for why this occurs. With unsupervised classification such as this, there are inevitably some unjustifiable results.

4.2.2 Comparison of BERT Models

This section compares the results generated with different hyperparameters. In this case, two language models fine-tuned using the methods described above are compared against two other, less trained, models. The models will be denoted as:

- A: DistilBERT model with no fine-tuning
- B: DistilBERT model trained by Amgen on deviation data
- C: DistilBERT model trained on causal factor data

- D: Deviation-trained model trained on causal factor data

In order to compare the models directly against each other, they are each used to embed causal factor text and the resulting embeddings are clustered using identical parameters. The following results are based on those clustering runs.

Table 4.1 shows a summary of all the quantitative metrics calculated for the cluster runs using each of the four models. Looking first at the percent of noise, we can see that model A has significantly more noise than any other model with around half of all causal factors going unassigned. Models C and D have the least noise and significantly less than model A. The number of clusters are all fairly similar. Clusters range in size from around 100 to 150 causal factors per cluster. Continuing in Table 4.1, we turn to 5M entropy. Here, we present the average 5M entropy which is calculated across all clusters within the cluster run. As a frame of reference, when requiring all clusters to contain a single 5M category, the average entropy values are around 2.3. We can consider this a maximum value. The minimum value for entropy is 0 which could occur if every cluster had a distribution of 5M categories identical to the reference distribution. Looking at the results, model A is again the worst of the four, and models C and D are far ahead. This indicates that the fine-tuned models did a better job of grouping causal factors with the same 5M category together. The final metric in this table shows the percent of deviations with a cluster diversity greater than 0.5. Deviations in this category have at least 1 cluster for every 2 causal factors. Here, all models are fairly comparable to each other though model B has the lowest value. It is possible that model B, being trained on deviations, is better at identifying and embedding based on words related to deviations rather than causal factors. It is worth noting that cluster diversity can be dependent on the number of causal factor clusters and the amount of noise from the cluster run making it difficult to interpret on its own.

Apart from quantitative measures, we can also visually assess the performance of the models. In Figure 4-4, we can see the causal factor embeddings projected into 2-dimensions. In the graph, each dot represents a single causal factor. From inspection, we find that model A produces embeddings that are the most condensed

Model	% Noise	# of Clusters	Average 5M Entropy	% of Deviations with Cluster Diversity >0.5
A	50.1%	67	0.645	80.5%
B	21.0%	84	0.712	74.7%
C	14.3%	71	1.09	85.3%
D	11.0%	73	1.05	83.0%

Table 4.1: Quantitative statistics describing clustering runs performed using various models to embed causal factor text. All clustering runs were performed with identical parameters apart from the model.

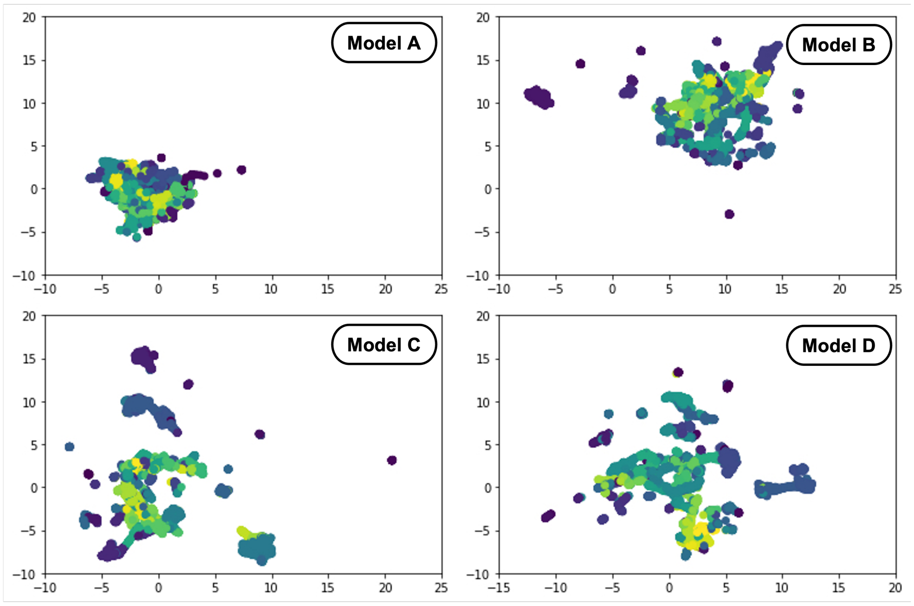


Figure 4-4: Comparison of 2-D projections of causal factor embeddings for four different language models. Each dot represents a single causal factor and the color of the dot indicates the cluster to which it is assigned.

in the smallest range. It is difficult to identify any clear clusters which likely explains the poor performance of model A across the board. B is more spread out with a clear set of causal factors distinguished in a group on the left though most causal factors remain in a more centralized area. Model C is more dispersed with additional clear groupings and model D produces embeddings that appear the most spread out of all the models and cover the largest range. These results further support the quantitative findings that the fine-tuned models (C and D) perform the best.

Next, we analyze the results from the frequency analysis of summary words extracted for each cluster within the clustering runs. Figure 4-5 displays two graphs:

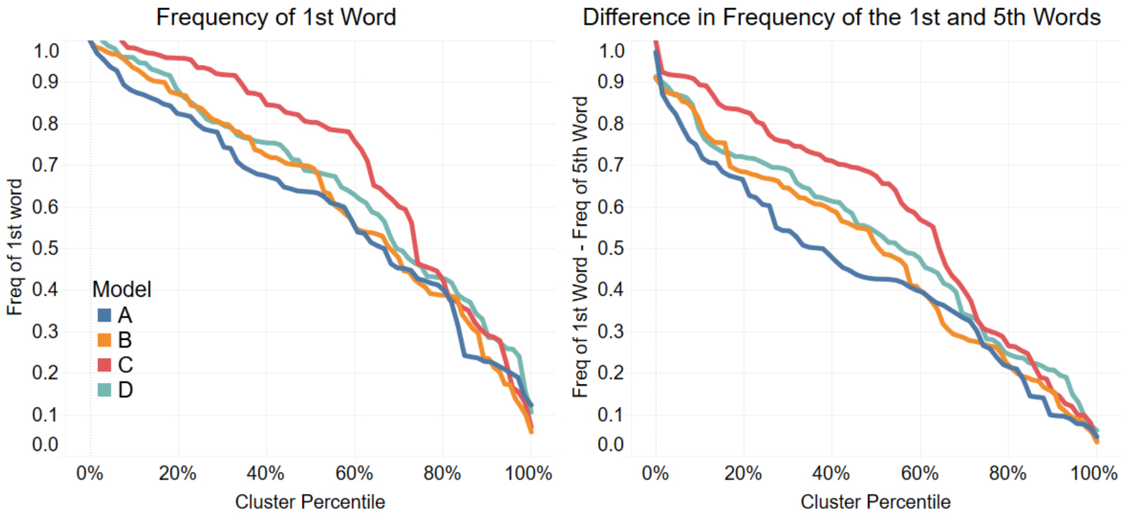


Figure 4-5: Comparison of summary word frequency metrics for clustering runs using four different language models.

the left is the frequency of the top, most frequent, word in each cluster and the right is the difference between the frequency of the top word and the frequency of the fifth top word for each cluster. The x-axis for both graphs is the percentile of the cluster as ranked by the metric. For example, reading from the left graph, the most common word in the cluster at the 40th percentile of clusters for model A appears in about 67% of causal factors in the cluster. Analyzing the results, we can see that model C is superior to the rest as its line appears above and to the right of the others in both graphs. The left graph indicates that more clusters from model C have a very highly common word than clusters from the other models. The right graph indicates that model C has more clusters with a large drop off between the frequency of the top and fifth words than others. In the right graph, we also see that model A is particularly poor indicating that the frequency of the top and fifth words are similar. See Figure A-2 in the Appendix for references graphs showing the spread of results using one model and identical parameters but different random seeds. This can serve as a baseline for expected variation between essentially identical runs.

Finally, using the summary words, we can calculate what percentage of the words are generic for each cluster run's cluster summaries. Generic words are less useful for investigators and thus, we want to limit their appearance in the cluster summaries.

Model	% Generic Summary Words
A	7.3%
B	4.4%
C	5.5%
D	5.3%

Table 4.2: Percent of generic words in cluster summaries for clustering runs for each of the four models.

The results are shown in Table 4.2. Model A has the highest percent of generic words though all are low enough to be acceptable.

Overall, the results clearly show that the models that were fine-tuned on the causal factor text are superior to the untrained models in regards to distinguishing the causal factors. Their embeddings produce cluster runs that have more clearly separated clusters, less noise, higher 5M separation, and more applicable cluster summary words.

4.2.3 Comparison of Text Cleaning

Next, we will compare the clustering results achieved from embedding text that has been processed in different ways. We will refer to each of these as different text "columns". All columns contain causal factor text of some kind but differ in how the text is prepossessed or what other fields are included. Four options are compared here and are denoted as:

- W: Causal factor text with all cleaning steps applied
- X: Causal factor text with the rationale text appended
- Y: Causal factor text without the causation filtering applied
- Z: Causal factor text without the codes translated

Overall, 21% of causal factors contain a causation word meaning that the W column does not equal the Y column. 11% of causal factors contain a code such that the W column does not equal the Z column. Therefore, we do not expect the results to change drastically when different text columns are embedded since a small, but not negligible, number of records are different. In order to effectively compare the performance of each

Column	Average % Noise	Standard Deviation % Noise
W	13.5%	2.3%
X	17.1%	0.53%
Y	15.9%	1.3%
Z	15.3%	1.9%

Table 4.3: Average and standard deviation of percent noise for clustering runs embedding four different text columns. Each row is an average of three runs with different random seeds.

column, we will use model C (DistilBERT model fine-tuned on causal factor text) and keep all parameters the same apart from the column being embedded. Additionally, we analyze the results over three runs with different random seeds to get a clearer view of how the choice of column to embed impacts the performance.

Table 4.3 displays the percent noise results averaged over the three runs. Here, we see that the fully processed causal factor column (W) has the lowest noise followed by the less processed columns (Y and Z). X produces the most consistent noise out of all the columns indicating that the runs are more stable and less impacted by changes to the random seed. This could be because X contains the longest string of text out of all the columns.

Moving on to qualitative assessments of performance, consider the distribution of keywords across clusters for each of the columns. Figures 4-6 and 4-7 show 2-D representations of these distributions. Here, each circle represents a cluster and the size of the circle is proportional to the number of causal factors in the cluster that contain the keyword. Overall, all columns do a sufficient job of grouping the keywords together. For "valve" in Figure 4-6, column W appears to be the most concentrated of the four. X seems to have "valve" in many different clusters and areas, and while Y and Z have some major clusters, the distribution is more spread out than for column W. For "train" in Figure 4-7, all columns seem to put the causal factors with "train" in one or a few clusters which is ideal and indicates that the model understands that this is a distinguishing keyword. For reference, Figures A-3 and A-4 in the Appendix show the baseline distribution of clusters and the distribution of a control word, respectively.

The final metrics we will consider are the summary word frequency statistics.

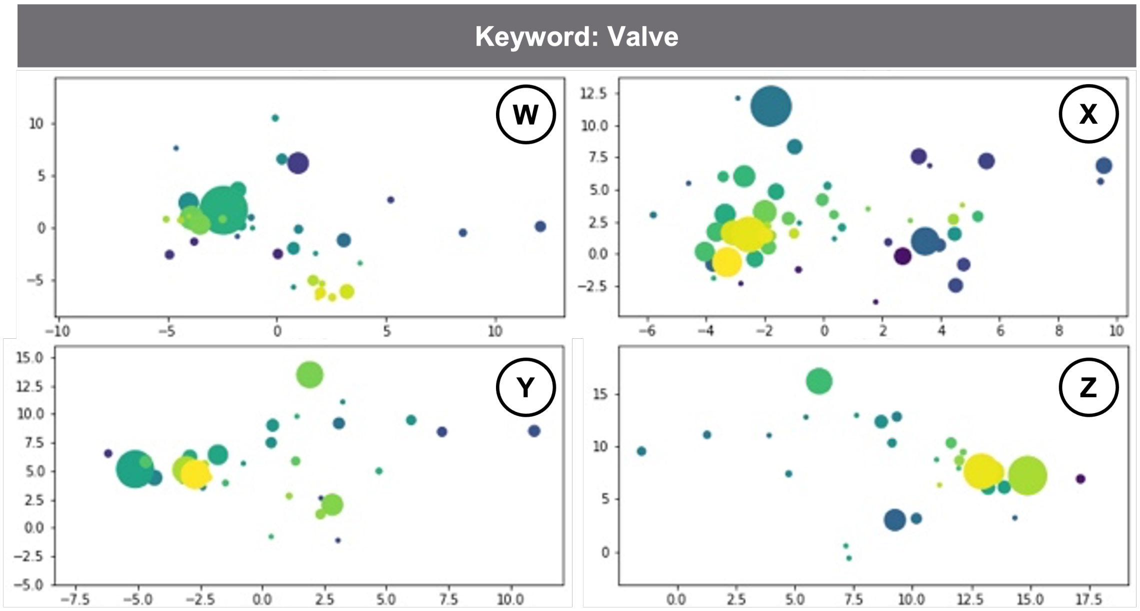


Figure 4-6: Comparison of the distribution of the keyword "valve" within causal factor clusters for four different embedded text column options.

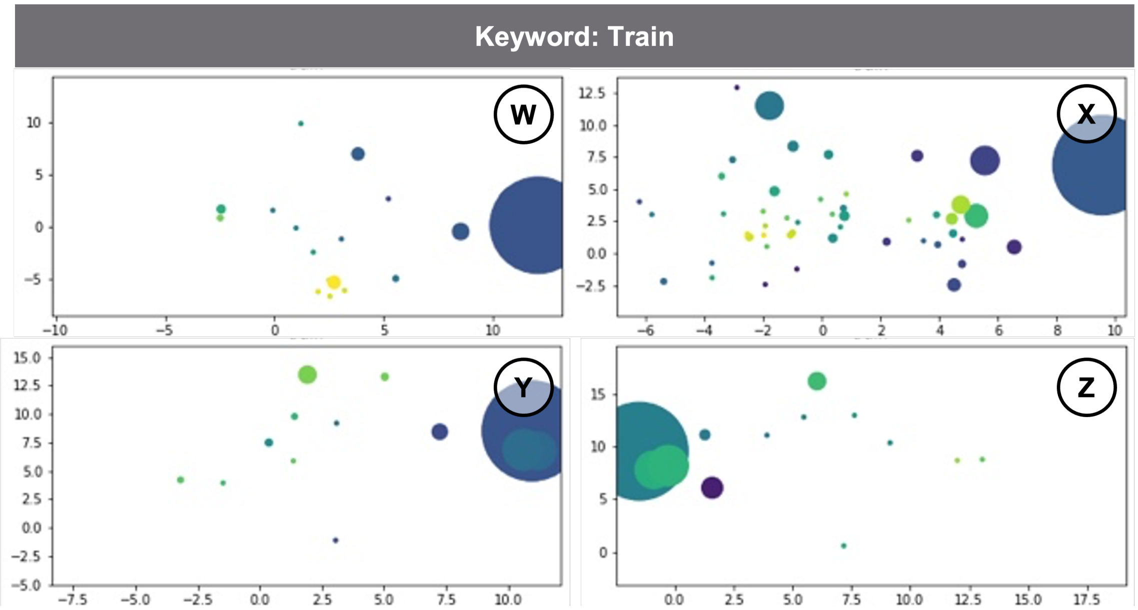


Figure 4-7: Comparison of the distribution of the keyword "train" within causal factor clusters for four different embedded text column options.

Figure 4-8 shows the comparison of column W (fully processed causal factor text) against the other three columns on the metric of top word frequency. Three lines per color representing one random run each are shown in the graphs. We can see that W outperforms X since all three runs generally appear to the right of all three X runs. W slightly outperforms Y and Z has the most variability. The similarity between W and Z is likely because, as presented earlier, only around 10% of causal factors have a difference between the columns. From the summary words, we can also calculate what percentage of words are generic for each cluster run's cluster summaries. The results show that column X has the lowest percentage (1.3%) while the rest have between 5.5-6.5% generic words. Likely, including the rationale text provides a larger number of words per record and increases the likelihood of finding a non-generic shared keyword.

In conclusion, we can see that all the options for columns produce fairly similar results using our metrics. It seems to be effective to include causation filtering and code translation since column W (all pre-processing) outperformed columns Y and Z (less pre-processing) on all our metrics, but the difference is very small. Including the rationale with the causal factor (column X) seems to create poorer performance on most metrics. However, there may be additional added benefits that warrant using this approach including generally providing the language model with more data and being able to find more non-generic keywords.

4.2.4 Comparison of Additional Parameters

Finally, we can look at the impact of two additional options in the clustering algorithm. First is clustering within 5M category which restricts the clustering algorithm to create clusters entirely within a single 5M category. In theory, causal factors in different 5M categories have different meanings. Clustering them separately could help improve the quality of the clusters. Another option is to separate by type of 5M category since some categories are more similar than others and may have some overlap. "Human Performance" and "Method" are grouped together as are "Materials" and "Machinery" while "Environment" is left on its own. We will compare the impacts of clustering by 5M category, clustering by 5M type, and clustering all records together by comparing

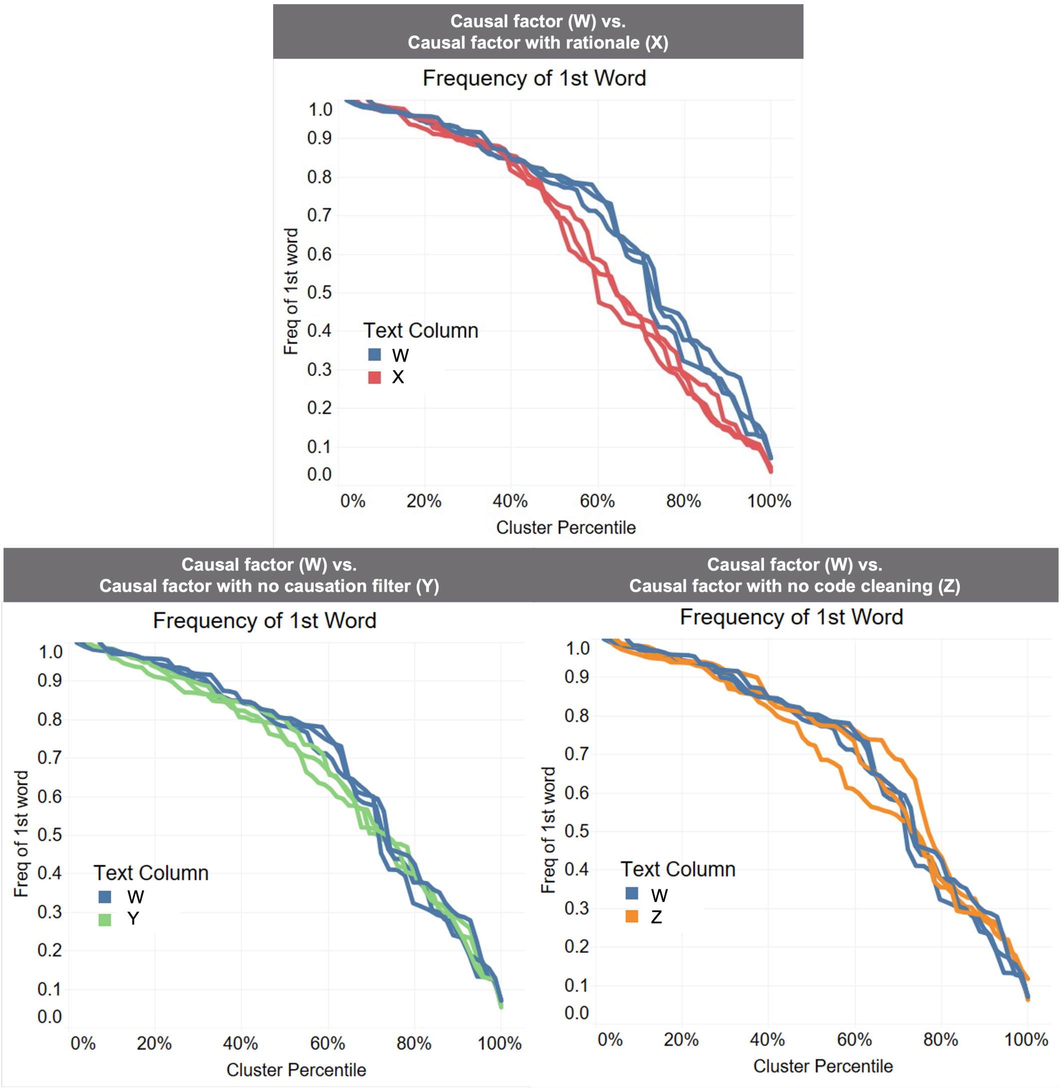


Figure 4-8: Comparison of summary word frequency metrics (frequency of top word) for clustering runs using four different embedded text column options.

Grouping	% Noise	# of Clusters	Average 5M entropy	% of Deviation with Cluster Diversity >0.5
All	14.3%	71	1.09	85.3%
by 5M	16.9%	92	2.32	93.2%
by 5M type	18.9%	91	1.98	90.7%

Table 4.4: Quantitative statistics describing clustering runs performed using different groupings.

cluster runs completed with model C (DistilBERT fine-tuned on causal factor data) embedding the causal factor text with all pre-processing.

Table 4.4 shows the quantitative metrics for the three schemes. From percent noise, we can see that the "no groupings" option (titled "all") produces the lowest noise. This is not unexpected. When clustering all together, there are likely causal factors in different 5M categories clustered together that when separated, do not have enough similar records to form a cluster and are therefore set as noise. There are also more clusters created when the groupings are enforced. Again, this is not unexpected since some clusters are similar across 5M category and would have been merged together if allowed. 5M entropy is not a useful metric for analyzing performance here since the clustering is constrained to group by 5M category, but the results are as expected: clustering within 5M category has the highest entropy, followed by clustering within 5M type, and clustering all together has the lowest of the three. Cluster diversity also isn't a very useful metric here. Nearly all deviations have a mix of causal factors in different 5M categories. Therefore, by definition, we would expect grouping by 5M category to have high cluster diversity. The results confirm this.

Next, we will look at the distribution of keywords within clusters as shown in Figures 4-9 and 4-10. Since all three grouping options use the same embeddings, the locations of the causal factors in the 2-D projection is identical. The only difference is how they are clustered. Looking at the "filter" example first in Figure 4-9, we can see that when clustered all together, there is one main cluster (the dark blue circle) that contains most of the causal factors with the word "filter". Moving to the 5M type grouping, we now see two equally sized large circles in that same spot. One likely contains the "filter" causal factors in "human performance" and "method" and

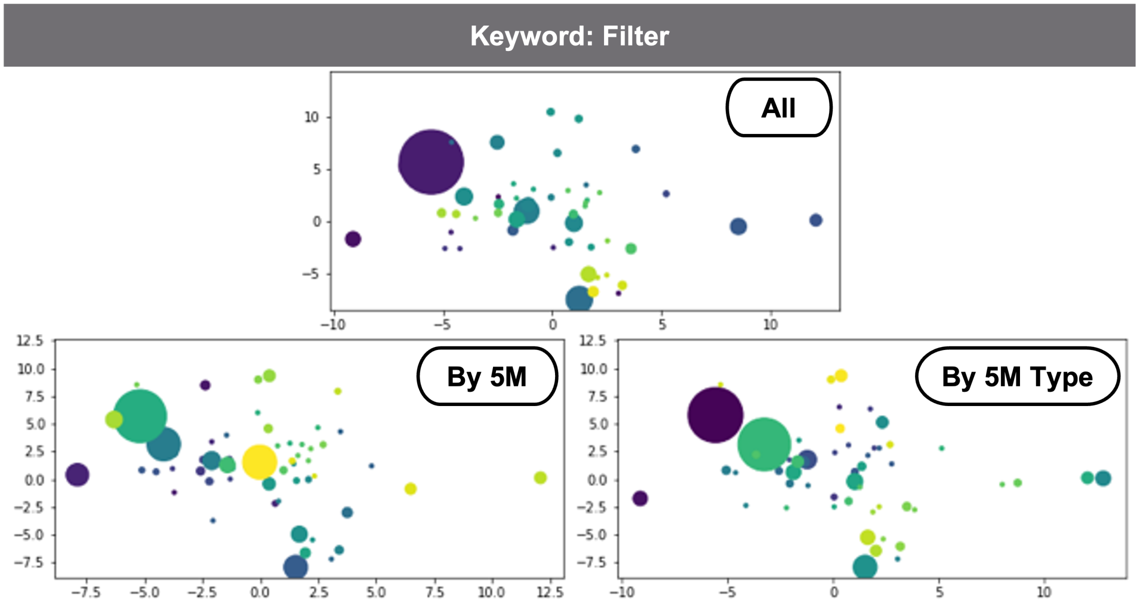


Figure 4-9: Comparison of the distribution of the keyword "filter" within causal factor clusters for three different ways of enforcing grouping in clusters.

the other, the causal factors in "materials" and "machinery". For the 5M category chart, the single cluster originally appearing when clustered all together is fractured even further. This happens because the keyword "filter" is not strongly associated with any particular 5M category so enforcing clustering within groups creates several similar clusters. In this case, it may be beneficial to have the large cluster broken down by category. Causal factors with "filter" in the "method" category could have a very different meaning from "filter" causal factors in "materials". Here, splitting up the cluster and enforcing separation by group likely creates clusters that are more specific. The "train" example in Figure 4-10 behaves very differently from the "filter" example. Here, regardless of grouping, there is one large cluster on the right side of the graph containing a majority of the "train" causal factors. This makes sense because the concept of "training" is fairly specific to the "human performance" 5M category. Organizing around 5M category therefore has no impact to this cluster.

Finally, we can compare the summary word frequency metrics shown in figure 4-11. Overall, we see that clustering with no groupings produces summary words that are more representative of the cluster than clustering within a group. This could be explained by the fact that clustering within a group creates a larger number of

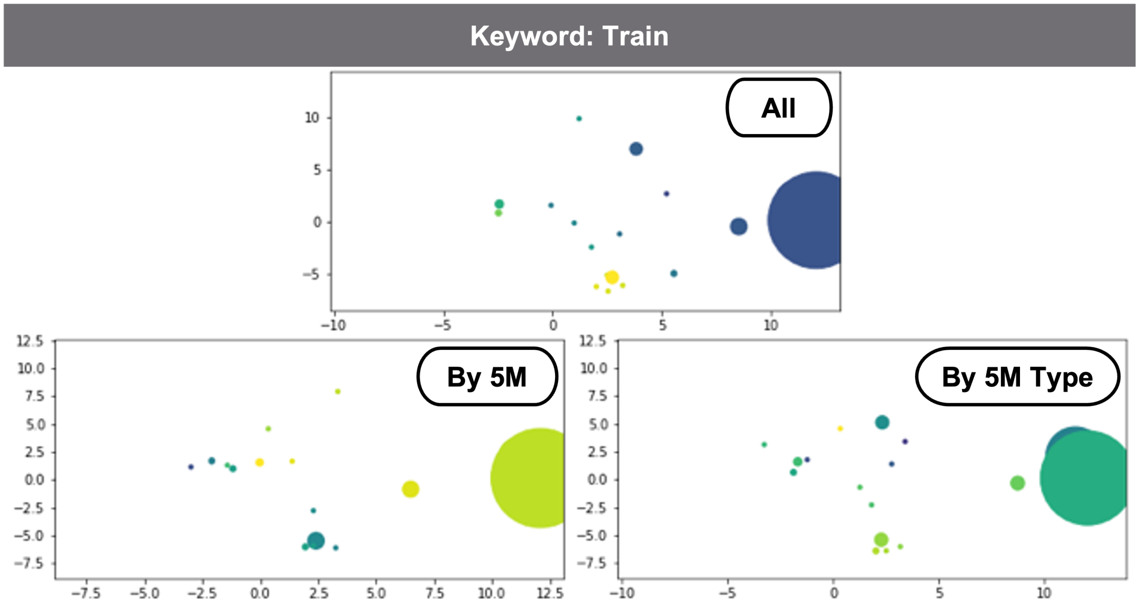


Figure 4-10: Comparison of the distribution of the keyword "train" within causal factor clusters for three different ways of enforcing grouping in clusters.

smaller clusters. Having even one misplaced causal factor in a small cluster can greatly reduce the frequency of the top word. In comparison, with larger clusters, having some misplaced causal factors is less likely to impact the top word frequency. Overall, the clustering run with no groupings produces the best results on our metrics. However, qualitatively there are some benefits to clustering within groups including breaking up large broad clusters into more meaningful smaller groups.

The final area of results we will consider is the impact of combining cluster runs created using identical parameters aside from the random seed. These combination clusters are created after the rest of the clustering algorithm is completed. The Methods section describes the process in detail. The goal of creating these combination clusters is to compensate for the randomness introduced by using UMAP to perform dimensionality reduction prior to clustering. Below, we compare the results of three random runs against the cluster assignments created by combining them. Results are shown using models C and D (the fine-tuned models), all with identical parameters apart from the random seed. The "Combo" cluster is created from "Rand 1" as the basis with "Rand 2" on top and then "Rand 3" on top of that so we expect it to be most similar to the "Rand 1" run.

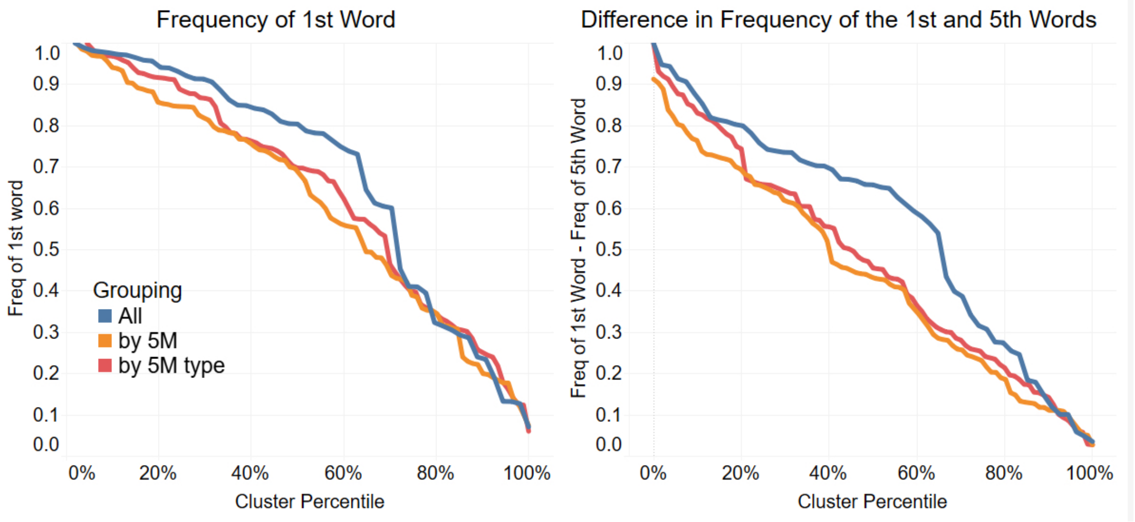


Figure 4-11: Comparison of summary word frequency metrics for clustering runs on the entire dataset, clustered within 5M category, and clustered within 5M type.

Table 4.5 contains all the relevant metrics that we will discuss. First, we can see that, for both models, the combination runs have less noise than the Rand 1 run. Though, for model D, the combination cluster has higher noise than both Rand2 and Rand3. As expected, the number of clusters for the combination runs is closest to their Rand1 runs. For cluster diversity, technically the combination runs are higher than Rand1 but not be a large amount. The percent of generic summary words decreases in the model D combination run but increases in model C so these results are not conclusive. Overall, the change in results using these metrics is very minor. It does seem like combining runs can reduce noise from the base run but otherwise has few impacts to our performance metrics.

4.3 Discussion

The previous section compares the outputs of the clustering pipeline using various parameters. Overall, some parameters produced higher quality results but no set of parameters is dominating. Generally, there are benefits and drawbacks to all options. In Section 4.2.2, which compares four language models trained in different ways, it is clear that the fine-tuned models do a much better job of embedding the causal factor

Model	Run	% Noise	# of Clusters	% of Deviations with Cluster Diversity >0.5	% Generic Summary Words
D	Combo	13.6%	89	83.6%	5.4%
	Rand 1	16.4%	90	83.2%	6.1%
	Rand 2	11.0%	73	83.0%	5.3%
	Rand 3	12.4%	79	83.9%	4.7%
C	Combo	10.0%	55	83.8%	6.4%
	Rand 1	10.9%	55	83.1%	5.8%
	Rand 2	14.3%	71	85.3%	5.5%
	Rand 3	15.3%	70	86.2%	4.9%

Table 4.5: Quantitative statistics comparing the performance of combination cluster runs against the original cluster runs they were generated from.

texts than the model with no fine-tuning. Even the model that was fine-tuned on deviations only rather than causal factors far outperforms the completely un-fine-tuned base DistilBERT model. This indicates the importance of fine-tuning the language model when the data uses highly specific and technical language. Next, Section 4.2.3 compared the results from embedding four texts with different levels of cleaning. Here, the outcome is less clear. Overall, it appears the causal factor text that includes code translation and the causation word filtering performs the best out of all the options tried, but not that significantly. It takes significant effort to translate the codes in the text and therefore, it may not be worth the small increase in performance. Section 4.2.4 beings by looking at the results when causal factors are first separated by 5M category and then clustered, as opposed to being clustered with no preliminary separation. We find that grouping by 5M category creates smaller more specific clusters that may do a better job of segmenting based on the overall meaning of the causal factor. However, this can potentially miss out on grouping together similar deviations across 5M category, which may lead to poorer performance in the case of summary words. Finally, the second part of Section 4.2.4 discusses how our method of combining clusters from several random runs in order to combat the randomness introduced by our dimensionality reduction software produced minor improvements. This method could be refined by starting with the clustering run with the lowest noise rather than an arbitrary cluster. There is some promise in the idea of combining runs, but the exact methodology needs to be developed. As it currently stands, it is likely an unnecessary step.

One of the most significant challenges in comparing the results of different cluster runs was determining how to assess the performance. In the absence of labels, we cannot use an accuracy assessment to decide which run is the best. Overall, the cluster evaluation methodology presented here provided some qualitative and quantitative ways of assessing the clustering performance but should not be assumed to completely capture and accurately assess the performance of the clustering. Particularly, the qualitative metrics that included 2D projections are highly subject to parameters used in dimensionality reduction meaning that these are not perfect comparisons. For the quantitative metrics, they can be impacted by other features of the clustering run such as amount of noise and number of clusters in a way that makes it difficult to compare across runs. The assessment methods presented here can only serve as a starting point for assessing clustering performance.

The basic methods discussed in this chapter can be easily applied in any environment where short documents of text need to be grouped together based on their topic. The basic flow of cleaning the data, embedding the text, and then clustering is not a novel idea. The methods used to extract keywords describing each cluster can also be used in any setting, but could be less effective if the text uses more synonyms. In our case, this method worked well because biomanufacturing language is fairly standardized. Most manufacturing environments are similar in that way and thus could use the same technique. The methods of data cleaning presented here such as code translation and causation word filtering are very specific to this dataset and thus not necessary for outside implementations unless they face similar issues.

Some of the metrics created for assessing cluster quality could be applied broadly while others could only be copied in concept. The qualitative method of graphing in 2-D and looking at the distributions of keywords is fairly generic and should work as long as there are relevant keywords in the dataset. 5M entropy is very specific to this dataset, but the same concept could be applied elsewhere if the data being clustered already contains some groupings that are relevant to the meaning of the data. The summary word frequency metrics would only be applicable if summary words were generated for each cluster. Overall, the method of unsupervised clustering and the metrics developed to assess performance are generally applicable across companies and industries.

There are many benefits and drawbacks to this unsupervised clustering methodology as a whole when considering how it could be implemented at Amgen. On the positive side, since this method is completely unsupervised, we do not need to tell the model what features of the text are important. The language model is able to determine this itself. It also means there is minimal upkeep in this method. Changes in the data will not be a problem as long as the language model is fine-tuned on some reasonable cadence. Since we are using a machine learning language model, synonyms and the meaning of the sentence as a whole are considered. For example, the model is able to understand that "filter testing procedure is insufficient" means the same as "filter test instructions are insufficient" which means the same as "inadequate instructions for performing filtering test". Simple statistical methods that do not use a machine learning language model may not understand that these sentence all have identical meanings. Finally, the methods described here are very similar to what Amgen is already doing to cluster deviations. This method would fit very well into existing processes and is already well understood and accepted by those who would be implementing it.

While there are many benefits, there are also some drawbacks. Fine-tuning the parameters used in embedding and clustering can take significant time and effort and it is difficult to know when you have found the "best" parameters. Additionally, changing the parameters can greatly impact the results in a way that sometimes seems unwarranted. For example, the fact that changing the random seed causes huge differences in the clustering output makes it feel less accurate and less like the cluster assignments are true reflections of the meaning of the causal factor. From an interpretability side, this method is a "black box model" such that it is difficult to explain to non-technical people why certain causal factors are grouped together and others are not. When we manually review the results from even the best clustering runs, we see that some clusters do not necessarily pick up on important distinguishing keywords. Often there are causal factors found that should not be grouped together and others in different clusters that have the same meaning. Looking at the cluster summaries, we see an overall inconsistent level of detail in the clustering outputs. Some cluster summaries may be very specific while other are broader. There is no way to control this. Apart from being inconsistent, the cluster summaries are sometimes completely wrong and do not represent their contents well. Many of these issues arise because of the sparsity of the data. Some causal factors are very common and can be clustered well while others appear very infrequently. If the minimum cluster size is too big, the less common causal factors will be missed, but if the minimum size is too small, there will be many similar clusters that should be combined. Overall, there are important concepts that the model might miss. There is no way to force the model to cluster around these known key concepts.

Chapter 5

Explicit Text Extraction

In Chapter 4, we saw that unsupervised clustering could be an effective way to group causal factors based on the meanings of their text descriptions. However, analysis of the results suggests that the model does not always cluster around topics that can be determined to be important and meaningful. Therefore, this chapter focuses on collecting and identifying a set list of words naming parts of the system that have been previously determined to be important in deviations, taking advantage of knowledge about the system to help classification. The results are used to draw interesting connections between items as they appear in the causal factors and deviations. The output is similar to that of unsupervised clustering in the sense that it allows us to group together records that are similar and analyze them as a group. However, in contrast to the output of Chapter 4, causal factors are not assigned to singular groupings. Instead, they are categorized based on the items they contain. This means that one record can be associated with zero or many groups, adding flexibility and allowing us to analyze causal factors in more than one dimension.

5.1 Methods

The focus of explicit text extraction is to identify words in the causal factor and deviation documents from a pre-set list of items. These pre-determined items are from four categories that are important aspects of causal factors and deviations:

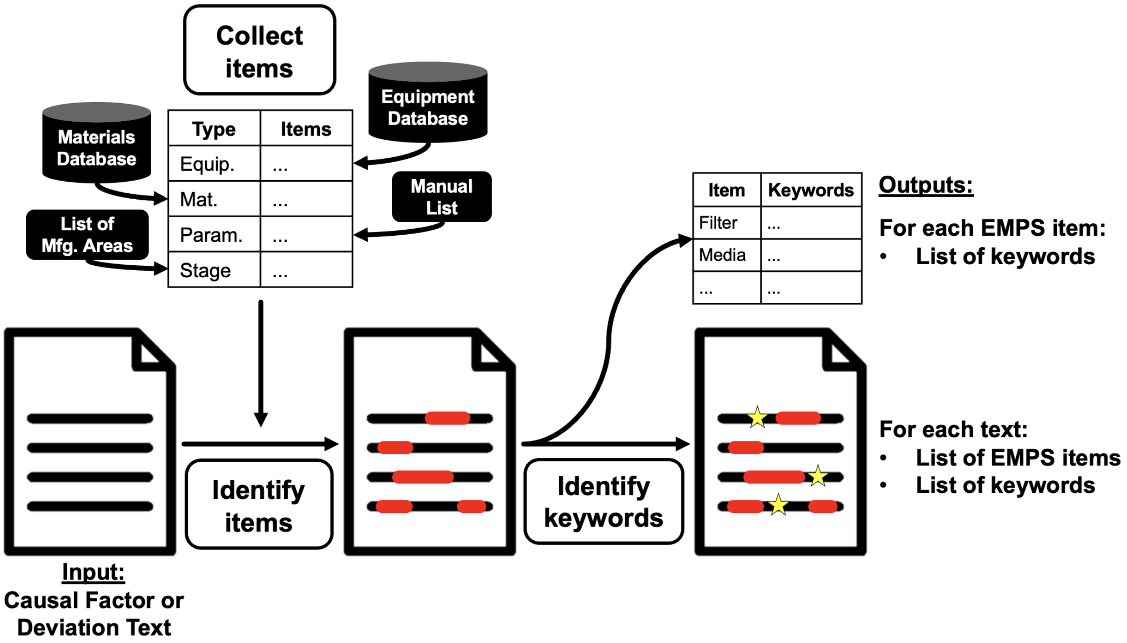


Figure 5-1: Overview of the explicit text extraction method.

- Equipment: for example, "bioreactor" and "filter"

- Materials: for example, "resin" and "media"

- Parameters: for example, "pH" and "total organic carbon"

- Stages: for example, "cell culture" and "purification"

We refer to these words as "EMPS items". Figure 5-1 provides a high-level summary of the steps involved in this method. First, a list of items is collected from a variety of sources. Then, those items are identified in the text which can be either the causal factor or deviation text. Finally, keywords related to each item are identified using a statistical method. The final output is both a list of keywords for each EMPS item as well as a list of EMPS items and keywords in each causal factor or deviation document. The results are visualized in network diagrams both to assess the quality of the output and draw insights into the system.

5.1.1 EMPS Identification

The first step in this process is to gather a list of EMPS items. Each type of item is sourced from a different place:

- Equipment: Sourced from the MAXIMO database which catalogs equipment and work orders. The list of all possible unique asset descriptions is used as the list of equipment names.
- Materials: Sourced from the Enterprise Data Fabric (EDF) materials database. The list of unique material types serves as the list of material names.
- Parameters: Created manually. A list of parameters was generated by identifying parameters mentioned in sources describing the manufacturing process steps.
- Stages: Sourced from the "area" field for deviations. The "area" field is used to organize deviations based on what high level step of the process they occurred in. The unique values from this field serve as the source of stage names.

In Chapter 3, we discuss these datasources in additional detail. Sourcing as much as possible from existing databases is the quickest way of generating a list of text items and avoids manual lists that may quickly become out-of-date. However, it is not possible in all cases.

Once the items are collected, they can be identified in the text. We will use two different methods to do this: text matching (for example, using the word "bioreactor") and code translation (for example, "01-R-1234" mentioned in the text is a code for a "bioreactor"). Codes are extracted and translated in a method similar to that described in the "Data Cleaning" section in Chapter 4. Regular expression text matching tools help identify and extract specific patterns of codes in the data. Particularly, we are looking for material numbers, asset and equipment numbers, work order numbers, and equipment codes in formats such as "00-XX-0000". The numbers are translated using the same datasources that the list of EMPS items come from. Asset and equipment numbers and work orders are translated using MAXIMO, materials numbers using the EDF materials database, and equipment codes from a manual file. The output of this step is a list of all identified codes in the text and their translations into plain text. Next, we identify the EMPS item directly in the text by looking for literal text matches. For each document, we iterate through all the EMPS items and use regular expressions to identify literal matches in the text. The matching algorithm includes the plural version of the word but no other inflected forms. Once this is complete, the list is combined with the translated codes to form a complete list of all EMPS items in the text.

5.1.2 Keyword Extraction

Once we have identified the EMPS items, we look for additional words appearing in the text that are related to those items to provide additional detail. Since the EMPS items are mostly nouns, we hope that these keywords will capture concepts such as adjectives describing a more specific subset of the item or modes of failure. To achieve this, we will look for words that appear commonly in the text surrounding each EMPS items. This based on the assumption that words related to an item are likely to appear in close proximity to it in the sentence.

The first step in identifying the keywords is to extract the words adjacent to the EMPS items appearing in the causal factor. For each EMPS item identified, we take the four words before and the four words after. If applicable, we truncate the adjacent text at the appearance of another EMPS item. This ensures no EMPS items are in the adjacent text of other items. Next, the adjacent text is lemmatized to remove stop words and set words to their un-inflected versions. Table 5.1 shows an example of how a possible causal factor "Issues in the bioreactor were due to incorrect installation of the valve and filter clogging" is split up into rows based on EMPS items and their adjacent text. The "Adjacent Text" and "Adjacent Lemmas" columns show the extraction before and after lemmatization.

The final result of the adjacent text extraction is a dataset consisting of one row per EMPS item identified in the text with a column containing the lemmas appearing around it. Using this, we can determine the related keywords by counting how many times a keyword appears adjacent to a specific EMPS item compared to how often it

Initial Text: Issues in the bioreactor were due to
incorrect installation of the valve and filter clogging

EMPS	Adjacent Text	Adjacent Lemmas
bioreactor	issues in the were due to incorrect	issue due incorrect
valve	incorrect installation of the and	incorrect install
filter	and clogging	clog

Table 5.1: Example of EMPS adjacent text extraction.

appears overall in adjacent text. This is similar in principle to TFIDF (text frequency inverse document frequency) but does not use the exact same formulas [10]. Keywords, here, are also allowed to be phrases rather than just single lemmas. There are four different types of lemma sets that are counted: "single" lemmas (one word), "pairs" of lemmas (two in a row), "skips" of lemmas (two words separated by one word), and "triples" of lemmas (three in a row).

To begin, all possible keywords are gathered from the dataset. This means we generate a list of all singles, pairs, skips, and triples of lemmas existing in the adjacent text. Then, for each possible keyword, we iterate through the unique EMPS items and count how many times it appears in adjacent text. We also count the number of times is appears overall and calculate various metrics based on the ratio of appearance next to the EMPS vs. appearance in general. From this dataset with one row per EMPS-possible-keyword combination, we filter to only keywords that appear more commonly near this EMPS than generally in adjacent text. For example, we would keep a word that appears in 50% of adjacent text to the EMPS item "flask" if it only appears in 10% of all adjacent text. We also filter the results using the following rules: the EMPS item must appear in more than 20 causal factors, the keyword must appear adjacent to the EMPS item more than twice, and more than 10% of the appearances of the keyword must be adjacent to this EMPS. In the end, we have a list of related keywords or phrases for the most common EMPS items.

5.1.3 Creating Network Diagrams

Once the EMPS items and their related keywords are identified, we can create network diagrams which allow the viewer to see relationships between items. One such diagram for our dataset might show the relationship between EMPS items appearing in deviations and the EMPS items appearing in those deviations' causal factors. These diagrams can become more complex by involving deviation clusters developed previously by Amgen and causal factor clusters as described in Chapter 4. We can also use data about the keywords for each EMPS item. There are many different ways of filtering and organizing the diagrams to show different and interesting connections. In the Results section, we will present various types of network diagrams that can be created from our dataset. The diagrams were generated using the pyvis Network package. All diagrams are interactive when presented digitally, so that users can drag the nodes and zoom in and out to have a better view of particular connections. It is also important to note that all network diagrams presented here have been filtered in order to make them more readable and only include strong relationships between items. These network diagrams provide interesting insights to user and also provide a way to assess and understand the outputs of the text extraction.

5.2 Results

In this section, we will look at the results of the explicit text extraction methods described above. This will help in determining the effectiveness of the method at describing causal factors and assisting in deviation investigations. We will begin by providing general statistics about the extraction. Then we move on to examples of network diagrams created from the results showing connections between deviation and causal EMPS items among others.

5.2.1 General Statistics

First, we will look at how many EMPS items we were able to identify. For text matching, we sourced around 90 material names from EDF Materials, Amgen's materials database, approximately 2000 equipment names from MAXIMO, Amgen's equipment database, and about 200 additional items from our manual list. Overall, we ended up with about 2400 items to search for in the text with most of those being equipment. For code matching, we had approximately 1300 unique translatable codes that were found in the text and matched in databases. This is out of around 2000 unique codes found in the text. Combining the results from code translation and text matching, around 600 distinct named EMPS items were found in causal factors and deviations. Of those 600, around three quarters are pieces of equipment.

The distribution of EMPS items found in the text is heavily skewed towards a few of the most common items. The top 20% of the most common EMPS items account for 87% of all occurrences of EMPS items in the entire text. Just the top 1% of EMPS items accounts for 27% of all occurrences. The most common item appears around 6000 times while the median is around 20. Some examples of the top few words include "bioreactor", "filter", "hood", "media", "manufacturing", and "tank". Additionally, text matches contribute a much larger number of the overall matches than the codes do. There are about 8 times the number of text matches for EMPS items than code matches.

Looking on a per record basis, we see that many documents (deviation description or causal factor description) have more than one EMPS item. The average number of unique EMPS items per deviation is 1.7. This goes up to 2 if we only consider deviations with at least 1 item. For causal factors, the average is 2.9 EMPS items or 3.3 items if the causal factor has at least 1. It is interesting to note that causal factors tend to have more EMPS items than deviations do. This could be because the causal factor text explored here includes the rationale text which may go into more detail around the investigation of the causal factor and may involve more parts of the system.

In Table 5.2, we can see how the different EMPS types are distributed within causal factors. Pieces of equipment are the most common EMPS items in causal factors with a majority of causal factors mentioning at least one. Stages also appear in most causal factors while materials and parameters are less common. There could be two reasons for this. First, our list of EMPS items may not capture all possible materials and parameters that are mentioned in the text. Secondly, these types are less applicable across all causal factors than equipment and stages. Overall, there are

Type	% of Causal Factors
Equipment	66.5%
Material	22.7%
Parameter	29.1%
Stage	51.1%

Table 5.2: Percent of causal factors containing each type of EMPS item.

no categories that are particularly uncommon such that they would warrant removal.

Other important metrics we are interested in are those comparing deviation EMPS items against their causal factor EMPS items. This is because connecting deviation and causal factor items allows us to answer the question: given a deviation with an EMPS item, what EMPS items should be proposed in causal factors? Further, we are more interested in cases where the causal factor mentions a different EMPS item than the deviation it belongs to. If the causal factor has the same EMPS item, it is likely just reiterating from the deviation. We are interested in causal EMPS items, not EMPS items referring to the deviation. For example, it would not be useful to tell an investigator if there is an issue with the "filter" to check the "filter". Table 5.3 shows the results of comparing deviation and causal EMPS items at the deviation level. In this table, all percentages are calculated as a percent of deviations rather than causal factors. As a reminder, there are many causal factors for each deviation. The table should be read from top to bottom. Each row adds an additional condition to the row above. For instance, the first row gives you the percentage of deviations that have an EMPS item while the second row shows how many have an EMPS item and have a causal factor with an EMPS item. Overall, the results are positive since the vast majority of deviations have an EMPS item and of those, nearly all have causal factors with EMPS items that are distinct. This indicates that our list of EMPS items covers a large number of deviations and that at least some causal factors per deviation have distinct EMPS items from the deviation. Table 5.4 shows similar results but at the level of causal factors. This means the percentages are calculated over the number of causal factors rather than deviations. Again, a vast majority of records contain at least one item and a high percent of causal factors contain an EMPS item that is

% of Deviations	
... that have an EMPS	83.1%
... and have a causal factor with an EMPS	82.5%
... and the causal factor EMPS is different	81.4%
... and the causal factor is confirmed*	62.9%
*89% of deviations have a confirmed causal factor	

Table 5.3: Comparison of EMPS items in deviations and causal factors at the deviation level. Each row adds another condition to the row above it.

% of Causal Factors	
... that have an EMPS	88.8%
... and belong to a deviation with an EMPS	75.4%
... and the deviation EMPS is different	69.6%
... and the causal factor is confirmed*	14.0%
*20% of causal factors are confirmed	

Table 5.4: Comparison of EMPS items in deviations and causal factors at the causal factor level. Each row adds another condition to the row above it.

different from the item in their deviation.

While our list of EMPS items seems to do a good job of covering a majority of records, there are still some that have no EMPS items. This could be because they mention something that is not on our list or that there are no applicable pieces of equipment, materials, parameters, or stages. For example, there are no applicable EMPS items for a causal factor such as "staff did not follow instructions properly". Table 5.5 shows the distribution of causal factors with no EMPS items across 5M categories. We can see those with none are concentrated in "Method", "Human Performance", and "Environment". This aligns with the idea that EMPS items as a concept may not be applicable for all causal factors since these categories tend to relate more to human aspects than equipment or materials. This is also evidenced by the very low percentage of causal factors with no EMPS items in the "Machinery" and "Material" categories.

Finally, we will look at the descriptive statistics for keyword identification. Overall, we identified an average of 50 related keywords for every EMPS item. Due to filtering, only the most common EMPS items have a set of identified keywords. Following are examples of keywords identified for a couple of EMPS items:

5M Category	% of Causal Factors with No EMPS Items
Machinery	3.6%
Material	7.4%
Human Performance	12.6%
Environment	12.9%
Method	15.4%
OVERALL	11.2%

Table 5.5: Percent of causal factors with no EMPS items by 5M category.

- Filter: "flush process integrity", "excessive pressure", "fouling"
- Agitator: "residue installation", "rotation", "speed"

In these examples, you can see that the keywords are very specific to the EMPS item in question. For example, "fouling" is a word describing the result of contamination building up on the surface of a membrane and is only applicable for filters. "Fouling" appears very infrequently in the corpus (<0.1% of records), so it was not identified as a keyword for any causal factor cluster generated with unsupervised clustering, but it is highly associated with the EMPS item "filter" which is how it was identified. This is an example where this method succeeds in finding important concepts that were too narrow for unsupervised clustering to identify. Overall, from manual inspection of the keyword results, there are many very good and highly applicable keywords for EMPS item. However, there are also plenty of keywords that are too generic or are highly repetitious. Because we look for singles, pairs, skips, and triples, there are sometimes very similar keywords for an item that all originate from a single phrase. For example, "dirty hold time" might be a triple that is highly associated with "tank". Likely, "dirty", "hold", "dirty hold", "hold time", and "dirty time" will all also be captured as keywords but they are unnecessary since their concepts are captured by the single keyword "dirty hold time".

Looking now at keyword matches within the text, we find that they occur in a large number of records. In total 86% of causal factors that have an EMPS item also have a relevant keyword for that EMPS item. Within those causal factors that have at least one keyword, the average number of keyword matches in a single causal factor is 13.4. This is likely inflated due to the duplicate keywords such as the "dirty hold time" example above. However, it is clear that these keywords appear commonly in the text.

5.2.2 Network Diagrams

Using the results of the explicit text extraction, network diagrams can be created connecting EMPS items in various schemes. Figure 5-2 shows example segments from one such network diagram. Here, red nodes are EMPS items appearing in deviations and the surrounding blue nodes are EMPS items appearing in their causal factors. Another way to explain this is that a red and blue node are connected if the EMPS item on a blue node appears in causal factors belonging to deviations with the EMPS item in the red node. In this case, the graph is filtered to include EMPS items from confirmed causal factors only. The first example segment on the top can be interpreted as saying deviations mentioning "filter" often have "filter housing" and "cartridge" in their confirmed causal factors. For a new deviation concerning a filter, these parts should be considered. The next two examples show more complex versions of the same deviation EMPS to causal EMPS relationship. On the left, "temperature alarm" is associated with a variety of related equipment pieces. On the right, we see an example where a parameter, "dissolved oxygen", is related to several pieces of equipment. Not all deviation EMPS items have a clear set of causal EMPS items, but it can be helpful in some situations. Adding and removing filtering conditions, for example, by filtering to a single area, can help focus the data and display clearer connections.

The next network diagram connects causal EMPS items to deviation primary clusters rather than the deviation EMPS items. These clusters may be a better way to group the deviations since they consider concepts outside of EMPS items and every deviation is guaranteed to be grouped in exactly one primary cluster. Similar to above, the diagrams can be interpreted as suggesting possible EMPS items to consider in causal factors. Therefore, it is useful that all deviations are guaranteed to belong to a primary cluster. Figure 5-3 shows three examples from this network diagram where red nodes are the primary clusters and blue nodes are the EMPS items appearing in

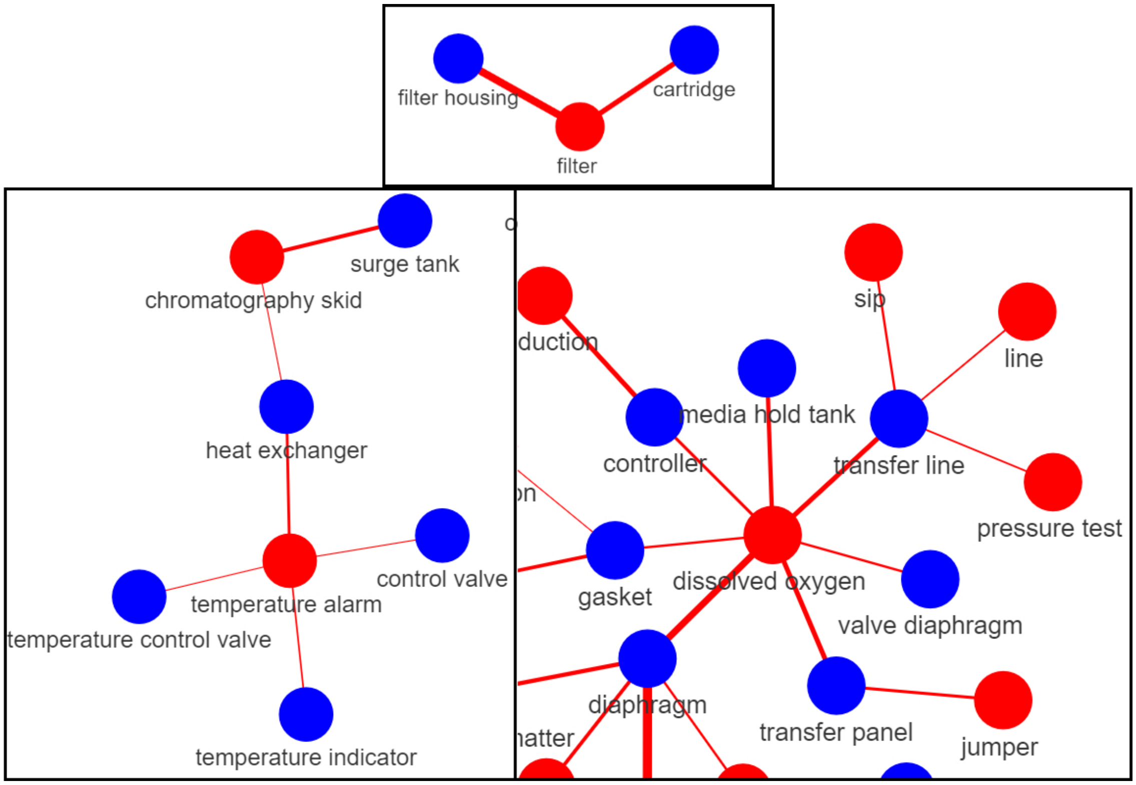


Figure 5-2: Examples from a network diagram connecting EMPS items appearing in deviations (red) to the EMPS items appearing in their confirmed causal factors (blue). The top graph can be interpreted as saying deviations with the item "filter" often have confirmed causal factors with the items "filter housing" and "cartridge".

their causal factors. An example of how to read the graph on the bottom is deviations in primary cluster 27 often have causal factors mentioning words such as "incubator", "scale", "flask", etc. The top left example centers on cluster 25 which has causal EMPS items all related to hood equipment and materials. The cohesiveness of the items indicates that this cluster and its causal factors have a clear topic. The example on the top right shows clusters 18 and 19 which have many overlapping causal EMPS items. This implies that these clusters are closely related. Finally, cluster 27 on the bottom is again related to a set of EMPS items of the same topic. Using a network diagram comparing deviation clusters to deviation EMPS item (rather than causal EMPS items), we find that cluster 27 is associated with EMPS items "viable cell density" and "shake flask". These items are highly associated with the causal EMPS items seen in the diagram such as "cell counter", "incubator", and "shake flask". Diagrams comparing deviation clusters to deviation EMPS items can help confirm that the relationships are valid, such as in this case. Those diagrams can also serve as a way of describing the deviation clusters.

Using the keywords identified for each EMPS item can provide additional context. Figure 5-4 displays examples from a network diagram connecting EMPS items (red) with some of their associated keywords (blue). Using the top graph as an example, we can read this as saying the EMPS item "probe" is related to the keyword "controlling" in causal factors. The diagram is filtered to only include keywords that are related to more than one EMPS item. This network is helpful in understanding how EMPS items may be related to each other. We would expect that EMPS items that are synonymous would have a similar set of keywords. The bottom image in Figure 5-4 confirms this thinking; "gasket" and "seal", which are similar pieces of equipment, share many keywords. On the top, the interactions are more complex. "Probe" shares keywords with "analyzer indicating transmitter" and "pH probe" which are both types of probes. An "analyzer indicating transmitter" may appear on a "incubator" which is why they share keywords, and "pH probe" is a probe that measures "pH" so they also share a keyword.

The final diagram will we present shows two levels of relationships and incorporates

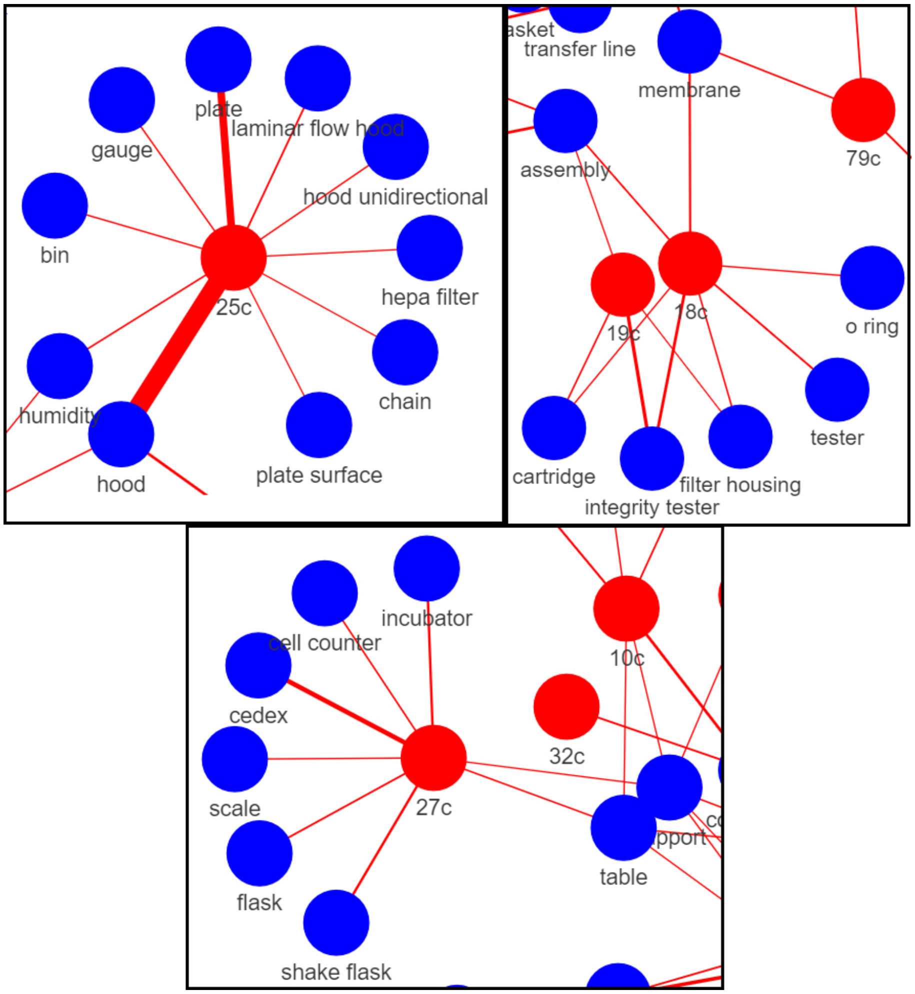


Figure 5-3: Examples from a network diagram connecting deviation clusters (red) to EMPS items appearing in causal factors in deviations in those clusters (blue). The bottom graph can be interpreted as saying deviations in primary cluster 27 often have causal factors mentioning words such as "incubator", "scale", "flask", etc.

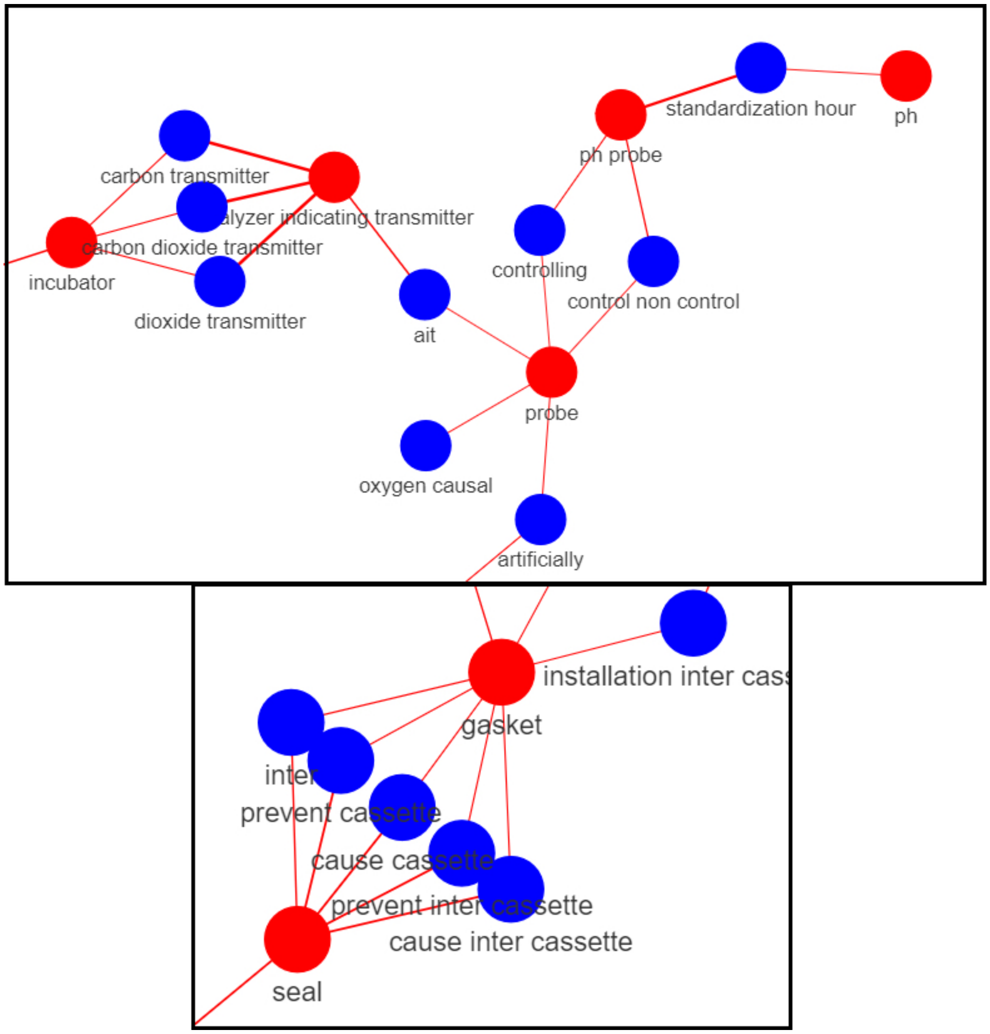


Figure 5-4: Examples from a network diagram connecting EMPS items (red) to the keywords associated them (blue). A portion of the top graph can be interpreted as saying the item "probe" has keywords including "controlling" and "artificially".

results generated using unsupervised clustering techniques described in Chapter 4. As a reminder, in that chapter we used machine learning models to group causal factor text into various groups or "clusters" based on their topic. Figure 5-5 has EMPS items as red nodes, causal factor clusters for causal factors with the EMPS item as the blue nodes, and keyword matches for the EMPS item appearing in those causal factors as the green nodes. Parts of the top left graph can be interpreted as saying causal factors containing the word "meter" when clustered in causal factor cluster 75 often contain the keyword "ultraviolet". The graphs in Figure 5-5 are intended to show that an EMPS item, when appearing in different causal factor clusters, may be related to different concepts. First, on the top left, is the word "meter" which when appearing in cluster 10 relates to "analyzer indicating transmitter" but in cluster 75 it relates to "ultraviolet". Here, the keywords serve as modifiers, clarifying the type of generic "meter". On the top right, "sample valve" is distributed across three different clusters. This is an example where the keywords relate to mechanisms of failure or actions taken with respect to the EMPS item. Finally, the bottom network for "weight" shows how the keywords can clarify the meaning of an ambiguous term. In cluster 15, weight is related to the concept of "weight indicating transmitters" which are a piece of equipment while in cluster 42 "weight" is used in the context of "high molecular weight" which is a phrase used to describe chemical compounds.

The networks described here are just a sample of the kind of diagrams that can be created using the data. Depending on the particular question being asked, diagrams can be made connecting any number of different pieces. The order in which the nodes are connecting can completely change the interpretation. For example, one could produce the same graph as in Figure 5-5 but flip the order of EMPS items and clusters to see for every cluster what EMPS items are contained and what their keywords are. Additionally, you can substitute causal factor clusters for deviation clusters in various settings. Network diagrams can provide powerful insights into the relationships between EMPS items and their distribution within clusters. The diagrams shown here indicate that our EMPS extraction methods can produce both expected and interesting results.

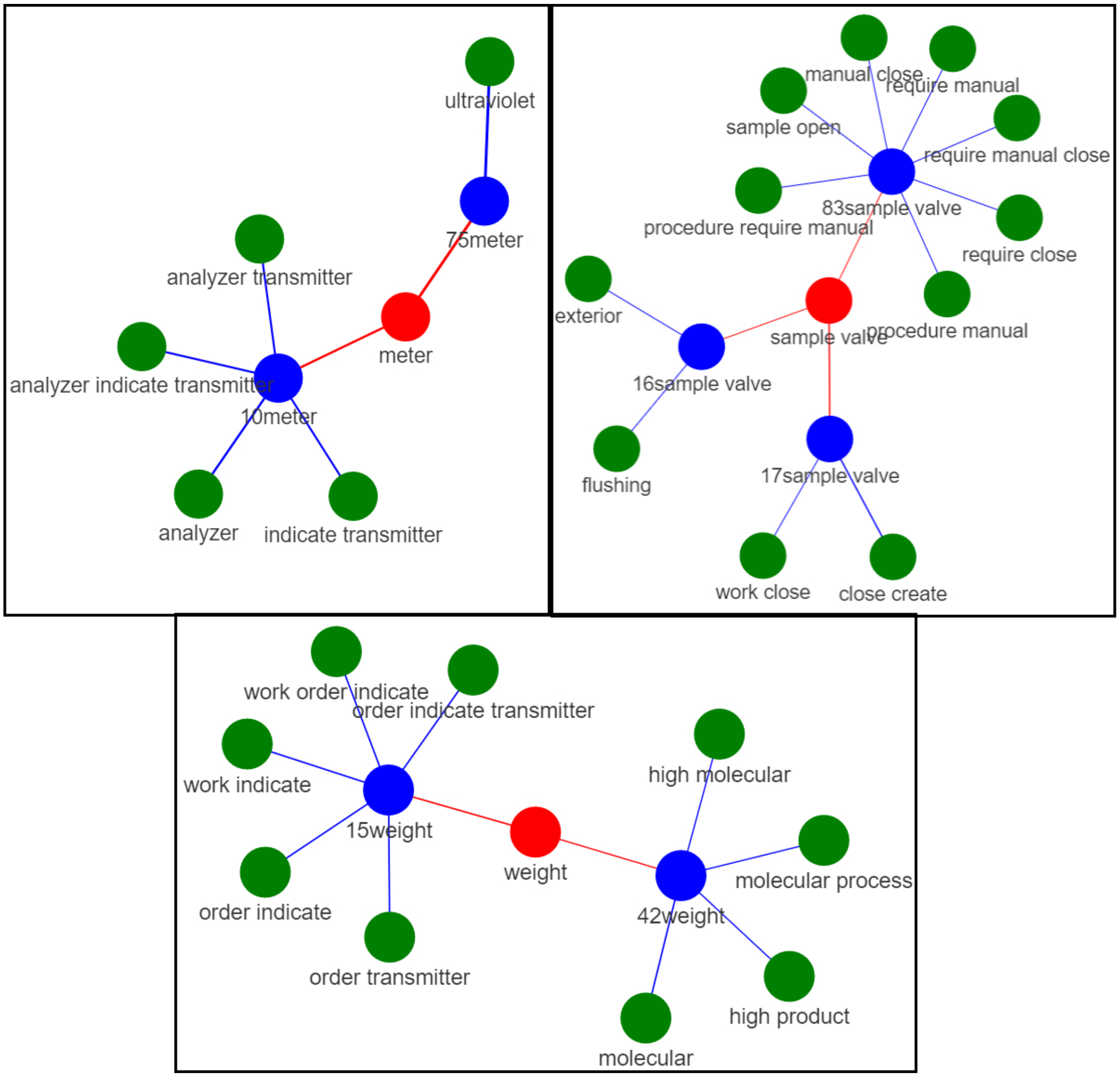


Figure 5-5: Examples from a network diagram connecting EMPS items in causal factors (red) to the causal factor clusters they are contained in (blue) and the keywords associated with the EMPS items in the clusters (green). A portion of the top left graph can be interpreted as saying when "meter" appears in a causal factor in causal factor cluster 75, it also often appears with the keyword "ultraviolet".

5.3 Discussion

From the general statistics regarding gathering and identifying EMPS items in the text, we can see that our methods were successful overall. A large number of potential EMPS items were gathered in an automated fashion by querying databases of equipment and materials. Some additional items were added manually including all of the parameters. In the future, Amgen should seek to automate this process as well. From the extraction portion, we saw that both deviations and causal factors have a large number of EMPS items, indicating that our list of items is fairly comprehensive. This is supported by data showing that the vast majority of "Machinery" and "Material" causal factors have an EMPS item. Those without tend to focus on topics where no potential EMPS item is relevant. While pieces of equipment make up the majority of potential EMPS items and those matched in the text, we still see that all four types are represented in a reasonable number of causal factors. When we compare EMPS items in deviations against those in their causal factors, we see that the pairs often contain different items. This indicates that causal factors propose new items that could be the cause of the deviation and do not simply reiterate the issue. For keyword identification, the process of identification worked well overall and succeeded in finding key concept words and phrases that were otherwise too infrequent for other methods to identify. However, the current method perhaps identifies too many keywords per EMPS item, many of which are duplicates. Additional parameter tuning and filtering of keywords could reduce these issues.

The results presented in network diagrams further support the success of the methodologies and provide interesting insights into how items are connected in deviations and causal factors. Such network diagrams could be used by investigators to find what kinds of items they should propose based on the EMPS items in their deviation or the primary cluster to which their deviation belongs. The keyword data provides additional context around the EMPS items and can serve to disambiguate generic or multi-meaning words or to describe different modes of failure. The network diagrams can make it easy to see connections, but sometimes the networks can become very complicated and cluttered, making them difficult to decipher. In those cases, other ways of viewing the data should be provided. Additionally, it is not clear if many of the results shown in these diagrams can provide novel and non-obvious connections to investigators. While the presentation is interesting, it might not continually provide value for an experienced investigator who might understand these connections intuitively.

The general methodology of explicit text extraction could be broadly generalized to any text data in any application, but many of the particularly useful strategies discussed here are specific to technical settings with a hierarchical text relationship such as deviation to causal factor. The steps that are generic are gathering a list of potentially important items, searching for matches to those in the text, and using statistical metrics to identify keywords related to those items. Theoretically, this strategy could help to classify pieces of text that use very specific language and where strategies like unsupervised clustering have fallen short. Applications with nested records can produce the added insight of drawing connections between items appearing in the parent and items appearing in the child. For our purposes, we have used "equipment", "materials", "parameters", and "stages" as the grouping of types of items. "Parameters" is a particularly important category for biomanufacturing but may not be relevant in other manufacturing settings. The rest are fairly generic and could be broadly applicable.

Overall, these text extraction methods have advantages and disadvantages in both their utility and in their implementation. On one hand, the methods use no machine learning and thus are easier for general users to understand. The outcome is also deterministic and the few parameters there are to tune have little impact on the final output. Particularly in the biomanufacturing space, having a deterministic output makes the work easier to defend. Additionally, by utilizing information about the system that we know is important for deviations (the EMPS items), we ensure that the output will be relevant and follow a particular structure. Finally, the results of this method have shown to be promising. Feedback from Amgen subject matter experts and others indicates high interest in the outputs, particularly the network diagrams.

On the other hand, this method is somewhat limiting. Without the use of deep learning models, we do not account for things like synonyms or inflected versions of words. Additionally, there is manual effort required in generated the list of EMPS items. Some of this can be automated by pulling from equipment and material databases, but there will always be some manual intervention to enhance the list. There is no way that we can generate a completely comprehensive list that covers every possible way of referring to an EMPS item. Over time, this list may also become obsolete and require updating. Finally, likely the biggest issue with this methodology is that it cannot consider the hierarchy of items. EMPS items can be connected to each in a multitude of ways including parts to subparts, type to subtype, or in a particular order in the system. Network diagrams can elucidate these connections, but they require human intervention to interpret. One example where this becomes problematic is in comparing deviation and causal equipment. It is not useful to suggest pieces of equipment that are larger or less specific than the equipment in the deviation. For example, if the deviation mentions a "filter" it is not useful to suggest "bioreactor" since the filter is a subpart of the bioreactor. Similarly, causal equipment should never be in a stage later than the deviation equipment. It would not be logical for an issue in the production bioreactor to be caused by purification equipment since this occurs much later in the process. Without a hierarchy of items, we cannot filter out these connections.

Chapter 6

Process Dependent Step Classification

Chapter 5 considered the identification of specific items in the deviation and causal factor text. Issues arose from the fact that the items were not connected in any kind of hierarchy or order. Sometimes connections were made between synonymous items, or a piece of equipment in a deviation would connect to a more general equipment in the causal factor. In response, this chapter investigates how additional knowledge about the manufacturing process as a whole can be taken into account to provide a structured output. This chapter presents an initial implementation using process data from standard operating procedure documents (SOPs). For the deviation data, we will use a subset of records with only one set of causal text per deviation, rather than multiple causal factors. The concluding section discusses the outcomes of this particular method as well as the general potential of using process information.

6.1 Methods

The goal of this method is to assign each deviation and its root cause to specific steps in the manufacturing process where they occurred. For example, we might find that a deviation occurred during the production phase of the bioreactor and its root cause occurred during media preparation. This assignment could provide interesting information about how quickly Amgen is able to catch problems as they occur. Ideally, the root cause should come from the same step or as close as possible as this indicates the issue was caught right away rather than allowed to propagate from an earlier step.

Our methods are tested on a dataset including major and minor deviations from only one site and for only one unit operation: the production bioreactor. This unit operation is one of the most important in the entire biomanufacturing process and is the final step in upstream manufacturing. Records have also been filtered to include only those with a root cause analysis. Not all minor deviations will include this. As mentioned in Chapter 3, this dataset contains less than 1,000 records.

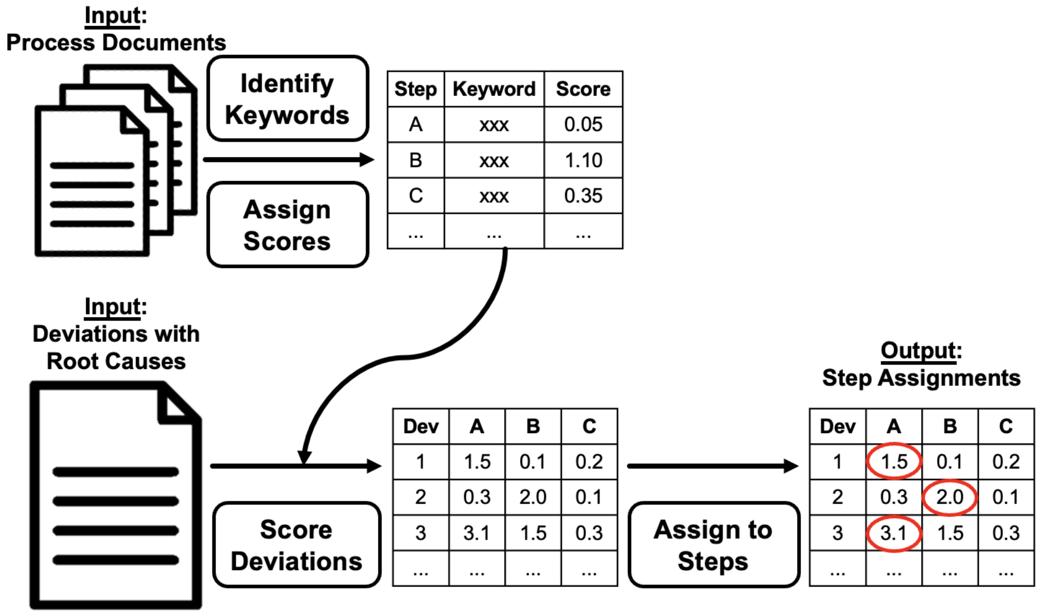
In addition to the deviations, we need a list of process steps. With the help of subject matter experts, we identified one SOP as the "Master SOP" which describes the process of running the unit operation from start to finish. The 8 main sections in this SOP are taken as the main process steps. Additionally, several other SOPs mentioned in the text of the Master SOP are also included as steps in the process and in our data, and are tagged with the Master SOP step that they are related to. Overall, this results in 8 Master SOP steps and 7 secondary SOPs for a total of 15 possible step assignments.

Figure 6-1 shows an overview of the step assignment process. We start the process by identifying keywords for each of the 15 process steps using the YAKE package. YAKE uses unsupervised statistical methods to extract keywords from single documents [4]. Given our process text, YAKE identifies 30 keywords up to three words in length for each step. We then assign a score to each keyword. A higher score indicates the keyword is more strongly associated with the step. The score is calculated from the formula:

$$score = \frac{\log(1+L) * \log(1+N)}{\exp{(1+M)}} \tag{6.1}$$

where L is the number of words in the keyword, N is the number of times the keyword appears in the relevant step, and M is the number of unique steps in which the keyword appears. The score is higher if the keyword is longer and if it appears more frequently in the step. The score is lower if the keyword appears across multiple steps.

Using the keyword scores, we assign a total score for each step to each deviation



Step	Keyword	Score
A	xxx	0.05
B	xxx	1.10
C	xxx	0.35
...	...	...

Dev	A	B	C
1	1.5	0.1	0.2
2	0.3	2.0	0.1
3	3.1	1.5	0.3
...	...	...	...

Dev	A	B	C
1	1.5	0.1	0.2
2	0.3	2.0	0.1
3	3.1	1.5	0.3
...	...	...	...

Figure 6-1: Overview of the process step assignment method.

and root cause text based on the words they contain. For the deviation, the short and long description fields are used, and for the root cause, the root cause analysis field as well as the several fields describing the CAPAs (corrective and preventative actions) are used. This approach is to increase the amount of text used for categorization. For each piece of text and step combination, the score given is a sum of the scores for the relevant keywords contained in the text. This raw score is converted to a percentage by taking the score for the step divided by the sum of all the step scores for the text. The percent scores for each piece of text are then used to assign the deviation and root cause analysis to their most likely process steps. Critically, the steps must be assigned such that the root cause occurs in the same step or earlier than the deviation. To achieve this, the deviation and root cause are jointly assigned to the pair of steps that maximizes the sum of the percentages subject to the ordering constraint.

In order to assess the performance of the assignment, a portion of the data (about 25%) has manually been labeled. This labeling only includes the deviation portion of the record and not the root cause. The labels can be compared to the assignments to generate an accuracy metric. Additionally, we can use mentions of SOPs in the text as a pseudo-label. We assume that, if a piece of text mentions an SOP, the text

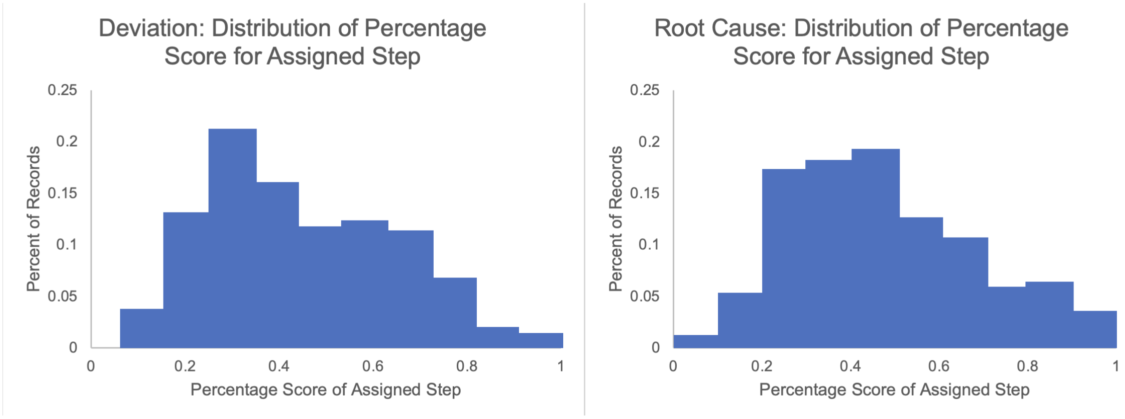


Figure 6-2: Histograms showing the distribution of percentage scores for deviation and causal factor step assignments.

should be assigned to that SOP. This generates another accuracy measurement.

6.2 Results

The results of the assignment are generally underwhelming. On the positive side, we see that 40% of deviations and root causes are assigned to the same step. This is as expected since, in many cases, the root cause and deviation text are very similar and occur in the same step. We also see that the assignments are fairly spread out over the possible steps. Some are more commonly assigned than others, but there are not one or two steps that dominate the rest. However, as shown in the histograms in Figure 6-2, the confidence of the assignments is very weak. The percentage scores shown in these histograms reflect the method's estimated percent likelihood that the record belongs to the step that it was assigned. The average for both deviations and root causes is around 45% and we can see that many are quite low, between 20 and 50%, while few are very confident, above 80%. This indicates that text generally contains keywords associated with many different steps making the assignment difficult.

Comparing the assignments against the manual step labels, we find an overall accuracy of 43% and a balanced accuracy (average of the accuracy of each class) of 53%. Notably, the maximum class appears in 64% of labeled data indicating that, if everything were assigned to the most common step, the accuracy would be

64%. Having an accuracy lower than the maximum class frequency indicates poor performance of the assignment algorithm. Looking at individual classes, we find that some are very accurate with performance around 75% while others get none or very few correct. We can get an alternate measure of accuracy using mentions of SOPs in the deviation and root cause text as pseudo-labels. We only use three of the most commonly mentioned SOPs that are also on our list. These give us "labels" for around 10% of text. The accuracy using the SOP labels is 81%, much higher than the accuracy using the manual labels. Notably, the manual labels generated were only for steps in the Master SOP while the SOP labels are for the secondary SOPs. The difference in accuracy could indicate that the assignment algorithm did a better job of assigning to secondary SOPs than the Master SOP steps. Additionally, it is possible that text mentioning an SOP is more likely to contain words from that SOP and thus is easier to assign.

For the dataset overall, we looked for mentions of any SOP, not just the ones in our assignment list, and found around 100 unique SOP numbers. 69% of records mention at least one SOP. This indicates that there is a much larger number of SOPs needed to describe the dataset than just the 15 Master SOP steps and secondary SOPs that we have used in this assignment. The discrepancy could explain the poor performance since many records do not belong to any of the process steps on our list. When we filter on records mentioning our SOPs, the performance is much better because those records fit into our defined process steps.

When we look at the keywords that are being found in the text, we find another issue that points to poor performance. After counting the number of times each YAKE keyword for the steps appears in the dataset, we found that the keywords with the highest scores are rarely or never matched in any text. Instead, the keywords that are less predictive are being matched, causing the assignments to be made using only these weaker connections. Using one process step's keywords as an example, the top 21 highest scored words (for example "filter oxygen isolation" and "bypass valve") account for only 1% of all the matches while the 9 other keywords (for example "method" and "filter") account for the other 99%. This leads us to believe that the classification is being based on somewhat arbitrary word occurrences.

6.3 Discussion

From the results shown above, it is clear that this method was ineffective at accurately matching deviations and root causes to process steps. The overall accuracy compared to manual labels was very low and the keyword matches determining the assignments were not meaningful. Looking at the data overall, we can see why this method is flawed. Firstly, there are many parts of the process that we are not capturing with the current list of process steps and would be difficult to add in. For example, tasks like routine cleaning of the facility are not covered in unit operation specific SOPs but can cause deviations. Additionally, there are some records that are not related to any kind of "process step" at all. For example, deviations caused by issues in supplier materials. There is a large variety in deviations which is difficult to capture. It is even difficult to manually label the records due to unclear wording of the text or the record not belonging to any clearly defined step.

The methods discussed here could be improved to compensate for some of the issues. For example, we could change the way that keywords are identified. Instead of using YAKE, we could use the techniques discussed in the previous chapter to look for pre-defined items in the text. This could help ensure that the keywords for each step are more meaningful. Additionally, we could try to remove records that do not correspond to any process step and categorize them separately. The root cause codes could assist in identifying root causes that are not process related such as supplier issues.

The overarching concept of using process information for text classification could be applied in any context with a well-defined process described in documentation. In particular, it is useful when the text being classified is, in a sense, "generated" from the documentation such as with deviation data. Given this, it is probably most relevant in manufacturing environments and for deviation and root cause text.

The idea of using process documents to classify deviations and their root causes into a linear process has significant promise. However, the exact implementation discussed here was not effective. We have already discussed why this methodology was flawed. Now, we will look at the advantages and disadvantages of process-driven classification for Amgen's deviations as a whole. On the positive side, this method takes advantage of known information about the system, including the ordering of events in the process and potentially a hierarchy of parts and materials. Being able to tag deviations and root causes to steps could help Amgen better understand causal relationships between different parts of the system. Additionally, the structured output (an assignment to a step) makes it easier to summarize the results at various levels, as compared to results in the form of text. Overall, using process data could help Amgen understand their deviation data at a new level and potentially take a step towards preventing future deviations.

However, there are some operational challenges with implementation. As mentioned in the Results section, there are around 100 standard operating procedures for one unit operation at one site. Expanding across all operations, we find around 1000 SOPs. It could be very difficult to categorize records into 1000 classes. Instead, we would need to group some SOPs together that are highly similar, remove SOPs that are irrelevant, and make other such editorial decisions. This manual effort to organize every SOP would be time-intensive and require constant updating since SOPs are regularly updated, added, and made obsolete. Further complications come from the fact that historical records will continue to reference old SOPs so we would need to maintain multiple versions of the SOP list. Additionally, this list of SOPs would only be the start of the documents needed to fully understand the entire system. There are many more document types used to map the pieces of equipment or sub-methods within the SOPs. Overall, it would be a large effort from Amgen's point-of-view to implement process-driven classification at scale. Currently, it is unclear exactly how impactful the results of such methods could be, making it difficult for Amgen to justify investing work into such a large project. More proof-of-concept implementations should be investigated to determine the feasibility and value more clearly.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Proof-of-Concept Tool for Investigators

In the previous three chapters, we discussed methods to draw insights from deviation and causal factor data using both machine learning and statistical techniques. Here, we show how the results from those methods can be disseminated to investigators through a reporting tool in order to help in the investigation of new deviations. We will begin with a baseline tool and then incrementally build on it using the outputs of the methods. Since our current implementation of process-dependent step assignment discussed in Chapter 6 was unsuccessful, it is not currently included in a potential tool. The proof-of-concept tool presented here was built in Tableau but could be recreated in any visualization software.

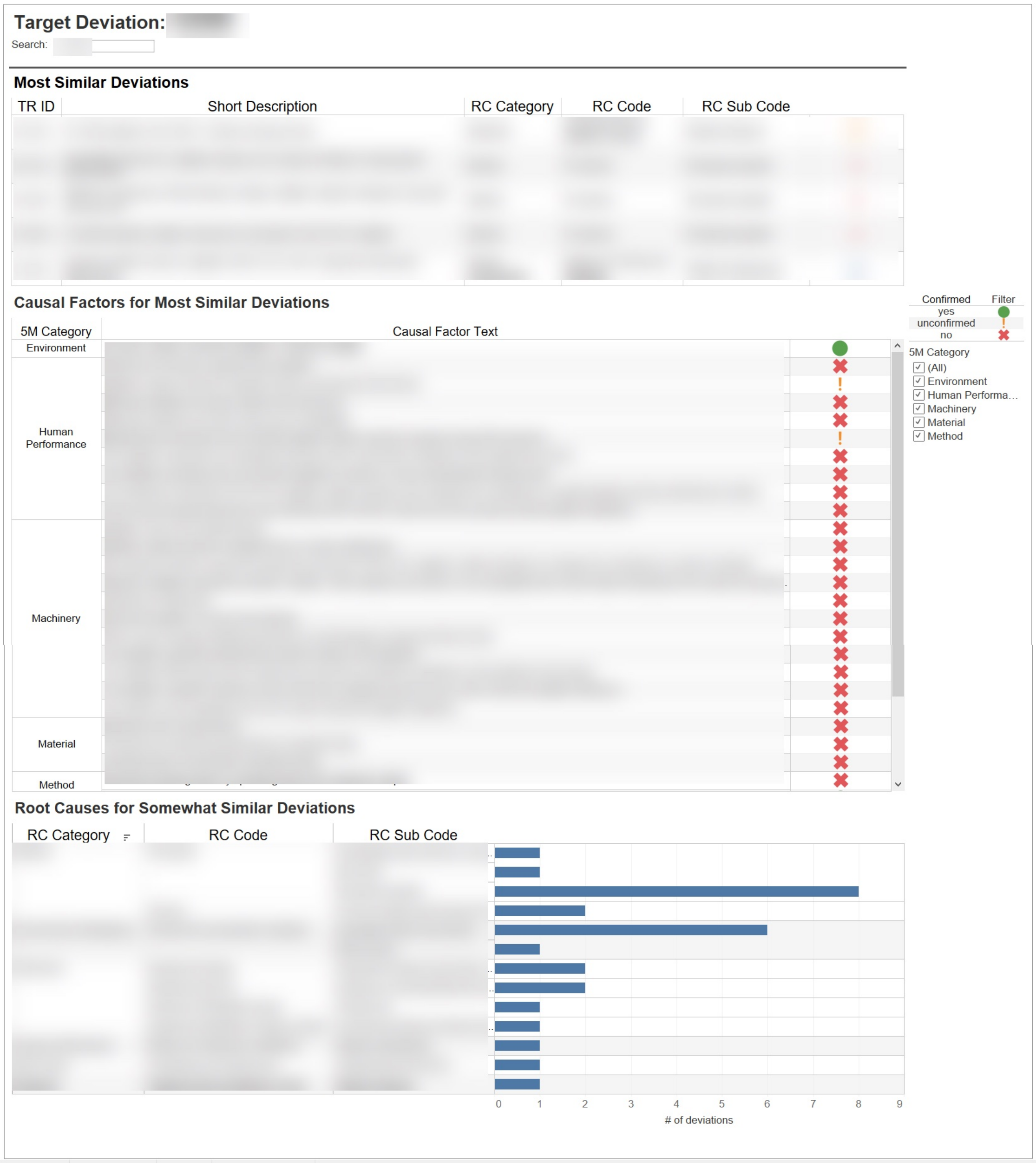
7.1 Baseline Tool

The baseline tool uses only data available to us at the start of the project with no transformations. Mainly, it uses the causal factor and deviation dataset as well the existing hierarchical clusters developed by Amgen which provide information about which deviations are similar to other deviations. For our purposes, deviations are considered "most similar" if they are grouped together in the narrower "secondary cluster" and "somewhat similar" if they are grouped together in the larger "primary cluster". Chapter 1 goes into additional detail about the existing Amgen deviation clustering implementation.

Figure 7-1 shows a redacted screenshot of the baseline tool. When a new deviation occurs, investigators can use the tool to find a list of similar deviations that have occurred in the past along with a list of all the causal factors that those deviations proposed. At the top, the user enters the ID number of their new deviation, and just below, a list of the most similar deviations appears with descriptive information including root cause category and root cause code. In the next section, the user is shown a list of the causal factors proposed in those deviations, organized by 5M category. An indicator on the right shows whether the causal factor was confirmed, ruled out, probable, or unconfirmed. The user can interact with the tool to filter causal factors based on 5M category, which deviation they belong to, and their confirmation status. At the bottom, the user is shown the breakdown of root cause codes used in deviations that are somewhat similar.

To know how effective this tool can be, it is important to know how many records (deviations and causal factors) an investigator would expect to see. Too many, and it could be infeasible to quickly read through, but too few, and it would not be useful. As a reminder from the data distributions shown in Chapter 3, there is an average of 22 causal factors per secondary cluster and 5-6 causal factors per 5M category within a secondary cluster. Looking at confirmed causal factors, we would expect the investigator to see about 5 per secondary cluster. Based on feedback from subject matter experts, these averages are within an acceptable range and should provide ample information for a new deviation to work from.

From an overall business perspective, this tool, even in the base form, has a large potential to help in the deviation process. Presenting the causal factors of similar deviations together in list form saves time for investigators who otherwise would manually search for historical reports and read through them for this information. Feedback from subject matter experts supported this thinking. However, this implementation is limited to looking at causal factors for highly similar deviations only. Beyond that, it is difficult to provide any kind of summary information.



Target Deviation: [redacted]

Search: [redacted]

Most Similar Deviations

TR ID	Short Description	RC Category	RC Code	RC Sub Code
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Causal Factors for Most Similar Deviations

5M Category	Causal Factor Text	
Environment	[redacted]	●
Human Performance	[redacted]	✖
	[redacted]	!
	[redacted]	✖
	[redacted]	✖
	[redacted]	!
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
Machinery	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
Material	[redacted]	✖
	[redacted]	✖
	[redacted]	✖
Method	[redacted]	✖

Confirmed	Filter
yes	●
unconfirmed	!
no	✖

5M Category

- ☑ (All)
- ☑ Environment
- ☑ Human Performa…
- ☑ Machinery
- ☑ Material
- ☑ Method

Root Causes for Somewhat Similar Deviations

RC Category	RC Code	RC Sub Code	# of deviations
[redacted]	[redacted]	[redacted]	1
		[redacted]	1
		[redacted]	8
	[redacted]	[redacted]	2
[redacted]	[redacted]	[redacted]	6
		[redacted]	1
[redacted]	[redacted]	[redacted]	2
	[redacted]	[redacted]	2
	[redacted]	[redacted]	1
	[redacted]	[redacted]	1
[redacted]	[redacted]	[redacted]	1
	[redacted]	[redacted]	1
[redacted]	[redacted]	[redacted]	1

Figure 7-1: Proof-of-concept Tableau dashboard

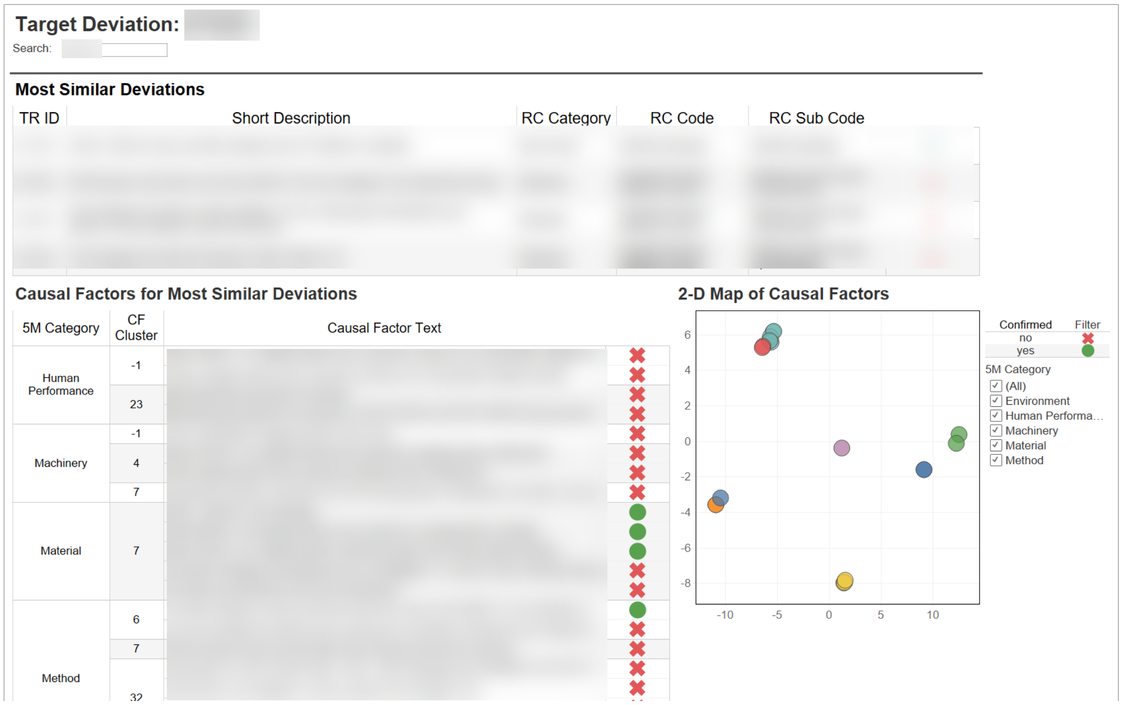
7.2 Tool with Unsupervised Clustering Output

On top of the baseline tool, we can add the output of the unsupervised clustering method discussed in Chapter 4. As a reminder, the output of this method is an assignment of every causal factor to a causal factor cluster. The causal factors are grouped based on the meaning of their text descriptions. Additionally, we have a few words summary of every cluster and a 2-D mapping of every piece of causal factor text. The 2-dimensional location of the causal factor relates to its meaning.

Figures 7-2 and 7-3 show screenshots of the updated tool. Now, the list of causal factors from highly similar deviations is grouped by causal factor cluster along with 5M category. In Figure 7-2, a graph on the right shows the 2-D projection of these causal factors, allowing for users to select an area on the graph and filter the list of causal factors.

Additionally, a new area of the tool seen in Figure 7-3 shows information about the causal factors for deviations within the same primary cluster as the new deviation. This gives the investigator a larger perspective on the causal factors that are commonly proposed. The visualization on the left lists all the causal factor clusters appearing in somewhat similar deviations to the target deviation. The length of the associated bar indicates how many times a causal factor from that cluster was proposed, colored by confirmation status. Along with the cluster number, the user can see the summary words associated with the cluster in order to understand what the cluster is about. On the right, there is a 2-D map of the center points of each causal factor cluster represented in the new deviation's somewhat similar deviations. The size of the circle indicates the number of relevant causal factors in the cluster. This mapping also allows users to select a causal factor cluster (or a set of clusters) and view a list of all the causal factors within them that appear in similar deviations. This provides an easy way to view causal factors at the primary cluster level; rather than all together, users can choose which clusters they want to see examples from.

The features added from the baseline tool give the investigator the ability to see more information at the primary cluster level that they could not see before due to



Target Deviation: [redacted]

Search: [redacted]

Most Similar Deviations

TR ID	Short Description	RC Category	RC Code	RC Sub Code
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Causal Factors for Most Similar Deviations

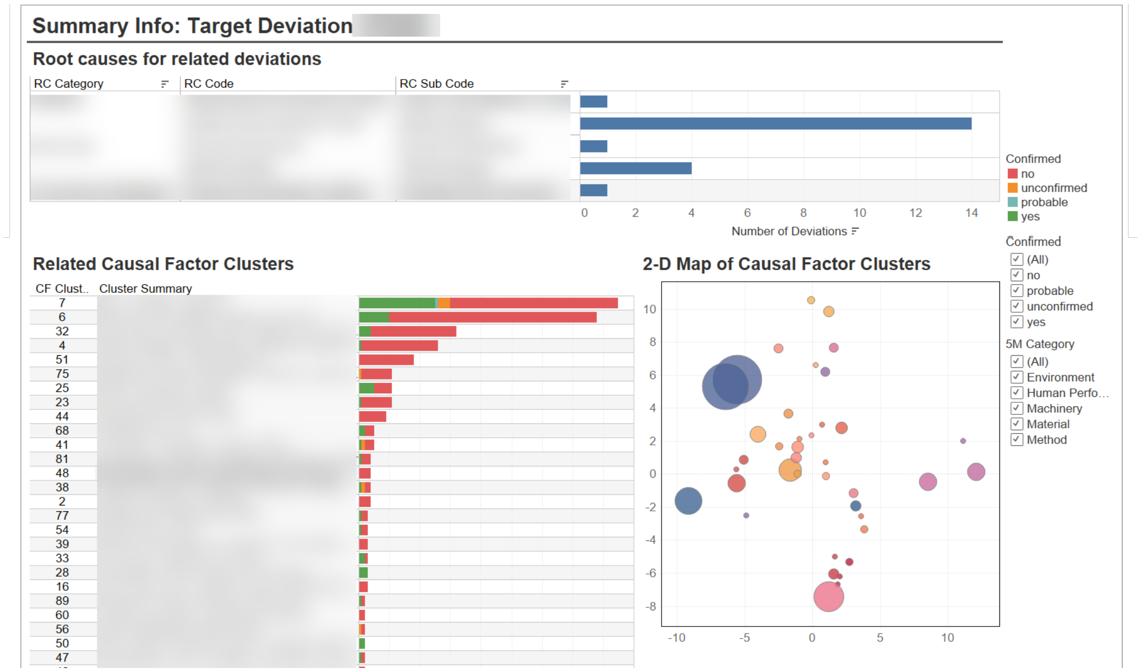
5M Category	CF Cluster	Causal Factor Text	
Human Performance	-1	[redacted]	✖ ✖
	23	[redacted]	✖ ✖
Machinery	-1	[redacted]	✖
	4	[redacted]	✖ ✖
	7	[redacted]	✖
Material	7	[redacted]	● ● ● ✖ ✖
Method	6	[redacted]	● ✖
	7	[redacted]	✖
	32	[redacted]	✖ ✖

2-D Map of Causal Factors

Confirmed | Filter
no ✖
yes ●

5M Category
☑ (All)
☑ Environment
☑ Human Performa...
☑ Machinery
☑ Material
☑ Method

Figure 7-2: Screenshot of proof-of-concept Tableau tool incorporating the output of unsupervised clustering. This is the first dashboard which focuses on causal factors for highly similar deviations.



Summary Info: Target Deviation [redacted]

Root causes for related deviations

RC Category	RC Code	RC Sub Code
[redacted]	[redacted]	[redacted]

Confirmed
🟥 no
🟧 unconfirmed
🟦 probable
🟩 yes

Confirmed
☑ (All)
☑ no
☑ probable
☑ unconfirmed
☑ yes

5M Category
☑ (All)
☑ Environment
☑ Human Perfo...
☑ Machinery
☑ Material
☑ Method

Related Causal Factor Clusters

CF Clust..	Cluster Summary
7	[redacted]
6	
32	
4	
51	
75	
25	
23	
44	
68	
41	
81	
48	
38	
2	
77	
54	
39	
33	
28	
16	
89	
60	
56	
50	
47	

2-D Map of Causal Factor Clusters

Figure 7-3: Screenshot of proof-of-concept Tableau tool incorporating the output of unsupervised clustering. This is the second dashboard which focuses on causal factors for somewhat similar deviations.

a lack of organization in the causal factors. The introduction of clusters of causal factors each with a cluster summary allows users to get a holistic sense of the type of causal factors they should be proposing for the new deviation. According to one investigator, the new segment of the tool showing data at the primary cluster level could be very helpful for the more "data-interested" investigators who are excited by these kinds of tools. However, it may not be useful to all investigators. The basic feature of presenting the causal factors for highly similar deviations is still the most useful overall.

7.3 Tool with Explicit Text Extraction Output

Finally, we can incorporate the output of the explicit text extraction method discussed in Chapter 5. The output of this method is a list of all EMPS items (equipment, materials, parameters, and stages) that are mentioned in each causal factor. On top of this, we have a list of keywords for the EMPS items appearing in each causal factor.

Screenshots of the updated tool are shown in Figures 7-4 and 7-5. The first dashboard shown in Figure 7-4 adds a list of all EMPS items appearing in the causal factors for highly similar deviations. This is presented as a bar chart with the length of the bar indicating the number of deviations that have a causal factor with this item. The color of the bar is based on the confirmation status of the causal factors containing the item. EMPS items in the bar chart are organized by EMPS type. This will give the investigator a sense of the items they might want to propose causal factors around.

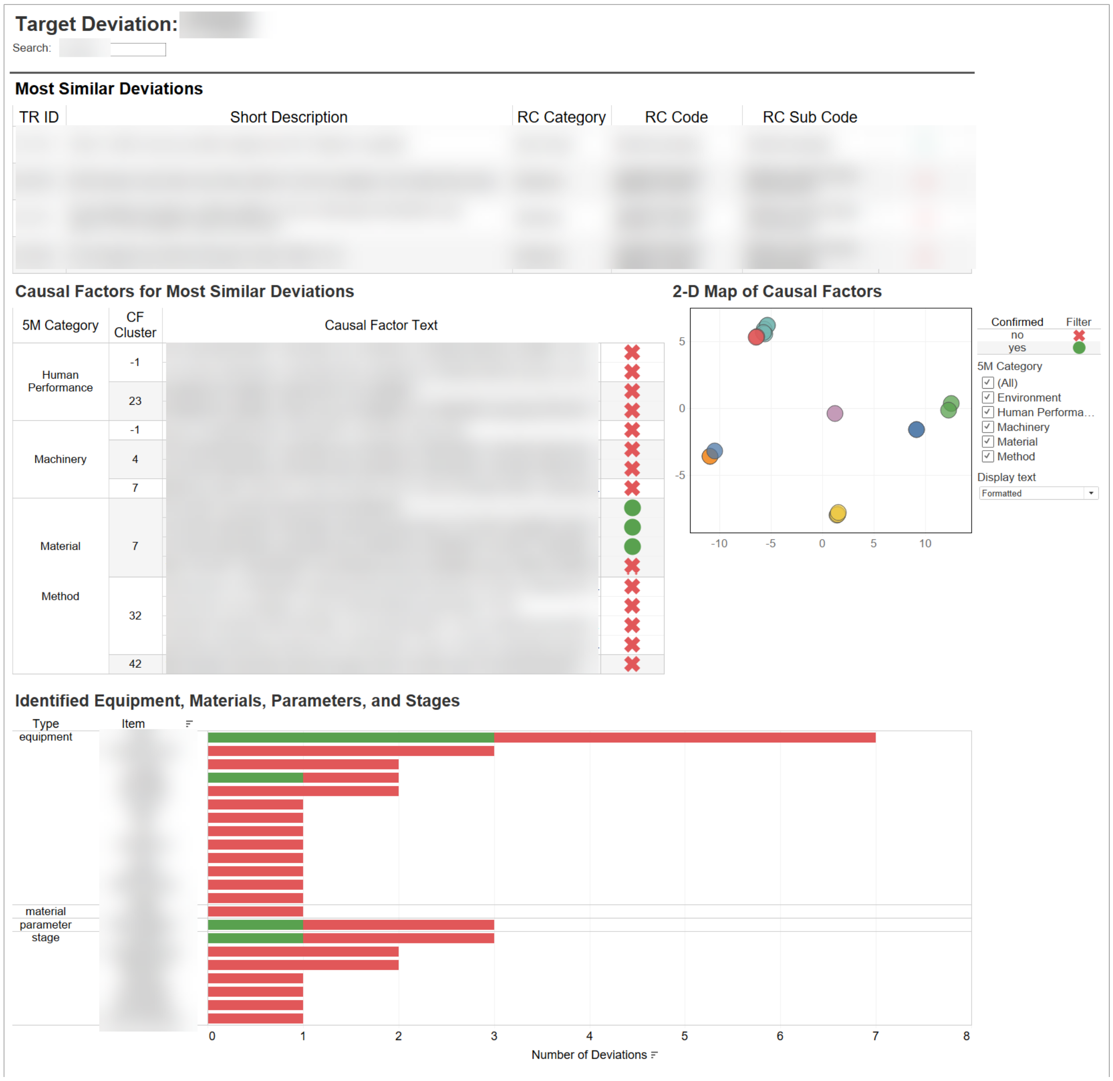
This dashboard also incorporates the keyword information by formatting the causal factor text shown in the middle section. This formatted version of the text highlights EMPS items as well as their related keywords in order to draw more attention to words that may be particularly relevant. A drop down selection on the side of the dashboard allows the user to switch between viewing the original unformatted text and the text highlighting EMPS items and related keywords. Since the contents of the tool are redacted in the screenshot, here are some examples of causal factors before and after formatting:

- Original: "Filter was damaged during handling", Formatted: "*FILTER* was DAMAGED during HANDLING"
- Original: "The slurry valve was installed incorrectly on 01-T-0123", Formatted: "The slurry *VALVE* was INSTALLED INCORRECTLY on *TANK*"

The words surrounded with stars are EMPS items and the capitalized words are the keywords.

The secondary dashboard shown in Figure 7-5, which contains information at the primary cluster level, has also been amended to include a listing of EMPS items in causal factors within deviations in the same primary cluster. The results are presented in a treemap where each rectangle is an EMPS item and the size relates to how often it appears. The four colors of rectangles are the four types of EMPS items. This provides an expanded list of the types of items that are usually proposed.

This most updated version of the report adds some new information on top of the previous version but likely is not required. The causal factor formatting highlighting EMPS items and their related keywords could be beneficial for some users particularly when there are many long causal factors to read through. The formatted text may make it easier to scan for keywords. The list of EMPS items is also a useful tool since it can give suggestions for EMPS items to include in causal factors, but the list is likely not critical for the success of this kind of tool since the investigator could get the same information from reading through each causal factor carefully. If possible, including some of the network diagrams described in Chapter 5 in the Tableau tool could provide additional benefits.



Target Deviation: [redacted]

Search: [redacted]

Most Similar Deviations

TR ID	Short Description	RC Category	RC Code	RC Sub Code
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Causal Factors for Most Similar Deviations

5M Category	CF Cluster	Causal Factor Text	
Human Performance	-1	[redacted]	✖
		[redacted]	✖
	23	[redacted]	✖
		[redacted]	✖
Machinery	-1	[redacted]	✖
	4	[redacted]	✖
		[redacted]	✖
	7	[redacted]	✖
Material	7	[redacted]	●
		[redacted]	●
		[redacted]	●
		[redacted]	✖
Method	32	[redacted]	✖
		[redacted]	✖
		[redacted]	✖
		[redacted]	✖
	42	[redacted]	✖

2-D Map of Causal Factors

Confirmed	Filter
no	✖
yes	●

5M Category
☑ (All)
☑ Environment
☑ Human Performa...
☑ Machinery
☑ Material
☑ Method

Display text
Formatted ▾

Identified Equipment, Materials, Parameters, and Stages

Figure 7-4: Screenshot of proof-of-concept Tableau tool incorporating the output of text extraction. This shows the first dashboard which focuses on causal factors for highly similar deviations.

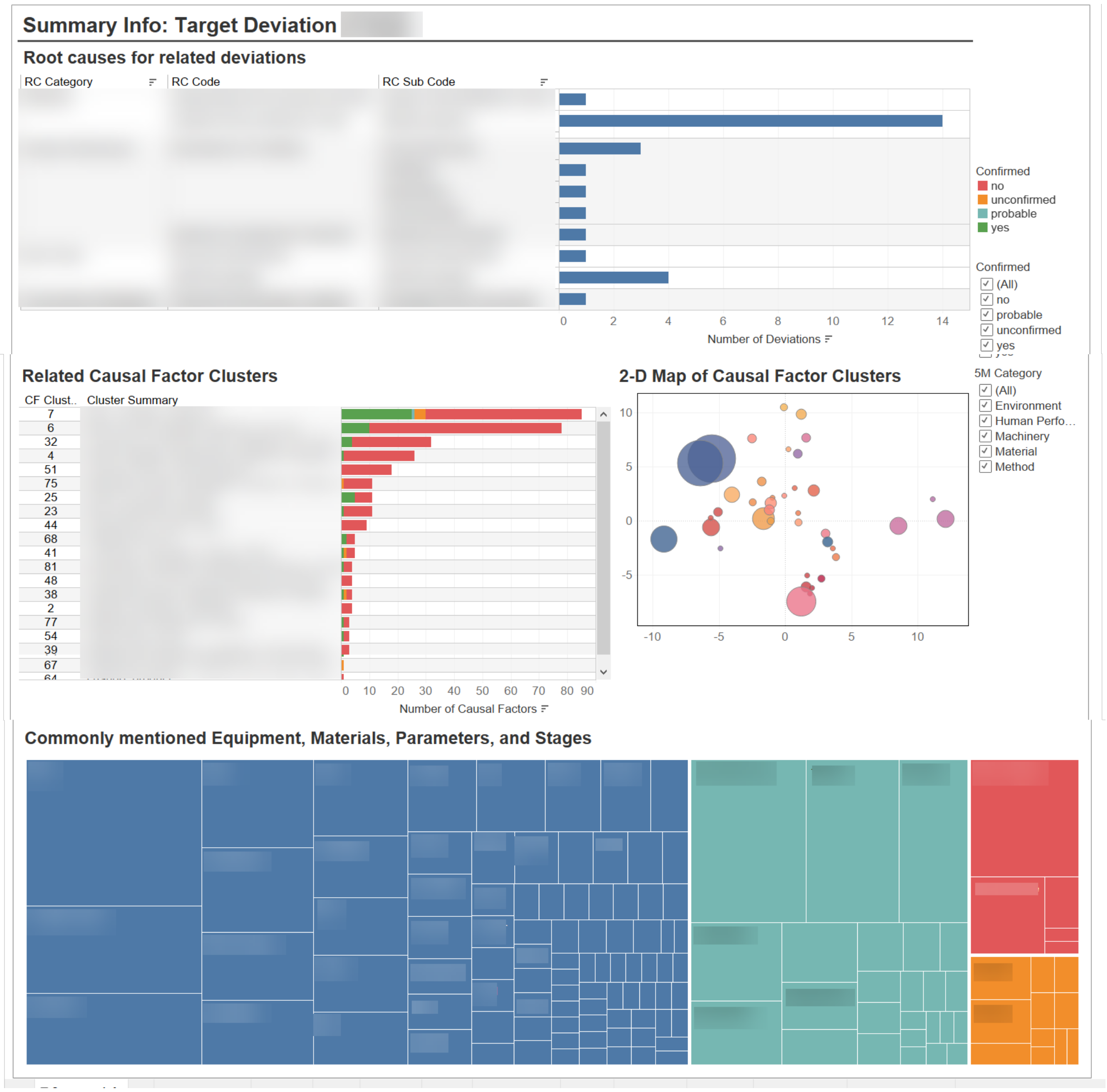


Figure 7-5: Screenshot of proof-of-concept Tableau tool incorporating the output of text extraction. This shows the secondary dashboard which focuses on causal factors for somewhat similar deviations.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Conclusion

In this thesis, we discussed the application of natural language processing tools in biomanufacturing particularly looking at Amgen's deviation investigations which are a critical aspect of ensuring quality. The goal of the project is to develop methods to extract insights from historical deviation reports focusing on text describing causal factors and the root cause analysis. Our dataset consists largely of free-text fields, so we used a variety of techniques including advanced Natural Language Processing (NLP) tactics. Here, we presented three such methods. First, we showed unsupervised clustering techniques which use machine learning language models to classify causal factors into groups. Explicit text extraction methods were implemented next to identify known key items in the text such as pieces of equipment and materials. Finally, we tried a process-driven approach using documents describing the manufacturing process to classify deviations and their root causes into particular steps in the process. Overall, the results were promising and offer Amgen several options moving forward. In this chapter, we will compare the methodologies in detail, describe how implementing some of these methods can be beneficial for Amgen, and provide recommendations.

8.1 Comparison of Methodologies

Each of the methods presented have benefits and disadvantages. Here, we will compare each of the methodologies on four key aspects: information provided, value to investigators, quality of results, and implementation. Amgen's decision in which path to take will depend on what value is the most important to them and how much implementation effort they are willing and able to dedicate to this project. Each method is not mutually exclusive and all of them could be implemented to help with deviation investigations, if feasible.

Starting by comparing the information provided, we see that each method has a distinct output. Unsupervised clustering provides groupings of causal factors based on their text as well as some summary words describing each grouping. The text extraction method gives a structured list of important items in causal factors as well as keywords related to those important items. Items and keywords can be used to group similar records together. Finally, the process-driven method provides an assignment of deviations and causal text to particular steps in the manufacturing process. No one output is more inherently more useful than any other.

Not only do the methods differ in their output, they also provide different value to investigators who are looking is use the results to inform investigations. The clustering method creates an understanding of the kinds of causal factors associated with similar deviations and allows causal factors to be summarized at a primary cluster level. The insights generated could not otherwise be gathered by investigators. The text extraction results help understand how specific items are related to each other and gives a structured output to investigators, namely a list of potential items to investigate. Lastly, the step assignments from the process-driven method help understand how problems in one part of a process lead to issues elsewhere. For a deviation in a given step, this could show which steps commonly contain the root cause. This method also has potential at a supervisory level to provide information about how long in process steps it generally takes to discover an issue.

Though all methods have the potential to add value, they were not all successful in achieving that potential. It is difficult to assess the quality of unsupervised clustering but, generally from the results presented, we see that the clustering algorithm did an acceptable job of finding distinguishing keywords and concepts. However, it is not successful in every case with some clusters being grouped together seemingly randomly. With this method, there is potential to miss out on key concepts. Using the text extraction methodology, we saw that most records contained an item in our list indicating that the list is fairly comprehensive. The network diagrams show interesting connections between items that are validated manually. Keywords extracted for items can be very specific but are sometimes hard to interpret or feel arbitrary. Finally, the process-driven results presented here were clearly unsuccessful in accurately assigning records to process steps. Different methodologies will need to be developed and assessed.

Lastly, we will compare the methodologies on ease of implementation for Amgen. Unsupervised clustering is fairly simple for Amgen to implement since it is very similar to Amgen's existing methods used to cluster deviations. Additional work will need to be done to fine-tune the model parameters. Text extraction methods are not particularly complicated to implement, but there is some manual work required to generate a complete list of items. As time goes on, Amgen will need to update this list to align with process and system changes. Process-driven methods, unlike the rest, may require a large amount of effort to implement and upkeep. As it stands, manual steps are needed to gather all the reference data and organize the process into discrete steps. This could serve as a significant barrier to large-scale implementation.

8.2 Potential Business Impacts

All of the methods used here have the potential to help with the efficiency and accuracy of deviation investigations which could have a major impact on Amgen's business. Deviations can lead to poor quality and potentially the loss of an entire batch. According to statistics gathered across biopharmaceutical manufacturing, batches can be lost at a rate of 7%. The impact of losing a single batch can sometimes cost more than $1 billion [16]. Most deviations are not so impactful, but just one problem leading to batch failure could be catastrophic. Therefore, it is critical for Amgen to maintain high-quality processes.

Based on feedback gathered from Amgen sources, the deviation investigation process can sometimes feel time-pressured. This means that investigators may propose system changes in the form of CAPAs that are less ambitious. The time pressure limits the ability to both generate and implement CAPAs which are the most important part of any deviation investigation since they are can reduce the likelihood and impact of future deviations. The methodologies described here have the potential to decrease the amount of time taken to propose causal factors which leaves more time to develop CAPAs and could lead to reduced deviations overall.

Beyond the application of these methods to particular investigations, there is potential to help understand the system as a whole. From the perspective of a manager or any person overseeing the manufacturing process, the data generated from unsupervised clustering, text-extraction, or process assignment could provide summary information about investigations. For example, the manager can see if there are any areas or pieces of equipment where deviations or root causes are common to occur and analyze trends in deviations and root causes. This could trigger improvements that may prevent recurring deviations.

8.3 Next Steps and Recommendations

The results and methods presented here serve as a proof-of-concept, demonstrating what could be done with the historical deviations dataset. Based on the quality of the results and feedback from potential users, we recommend that Amgen go forward with all the methods presented here in some capacity and to continue to make updates to improve upon the methods. Right away, we suggest that Amgen work to incorporate causal factor information into existing tools which present deviation trends. This would be very straightforward to implement and could instantly provide value to investigators. It would save the time of the investigator who otherwise would have had to lookup, download, and read-through each report for similar deviations to find the same information. In the medium-term, Amgen should work on finalizing the unsupervised clustering and text extraction methods. For unsupervised clustering, this includes training the language model on the full dataset and fine-tuning parameters.

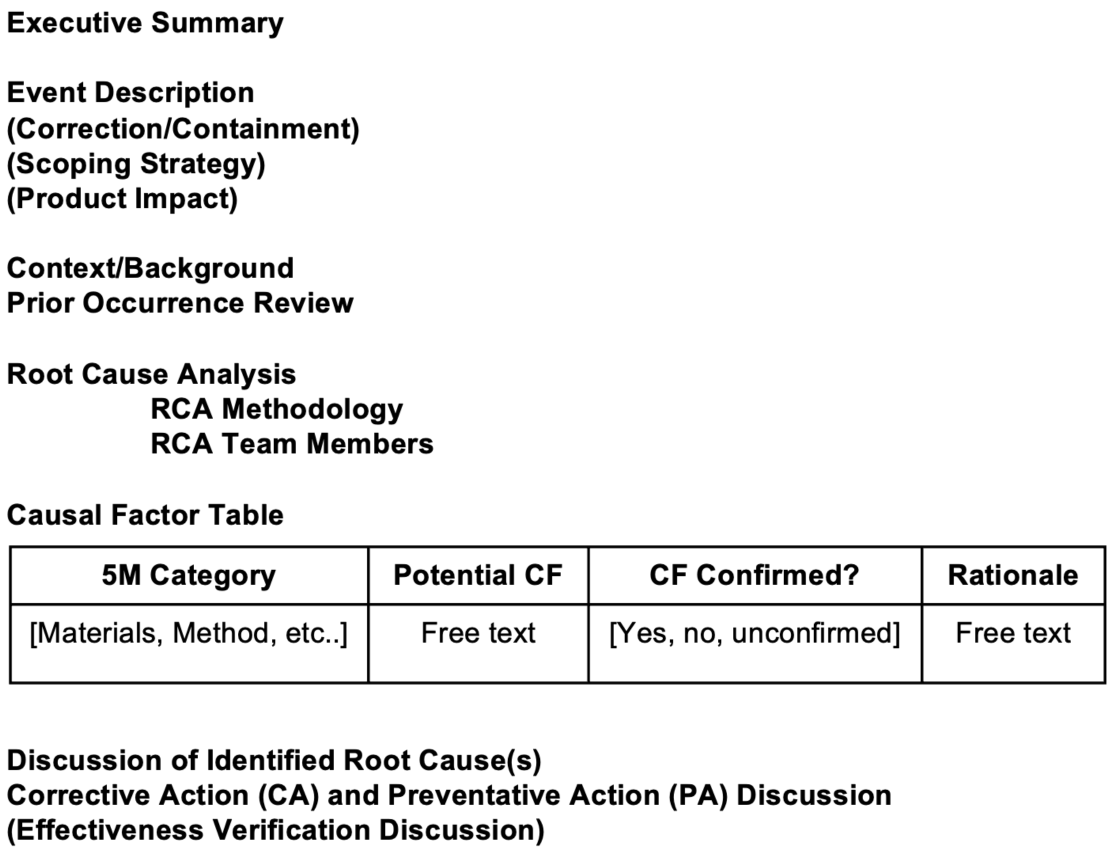
For text extraction, this includes expanding the list of items as needed and determining if there is a way to add any hierarchy to the list of items. The output of both these methods can be incorporated into the existing tool or in a new tool. Finally, in the long-term, Amgen can continue to investigate the potential of using process data to enhance the insights generated from the causal text. This will take more time to develop an effective method.

One of the key aspects to Amgen's success with any of these methods is the availability of the causal factor data. Work will need to be done to gather all the causal factors for historical deviations that have not already been extracted for this project. In the long run, Amgen should seek to change the method of capturing causal factors, entering them directly into a database rather than in a document file that must be processed to extract the information. This will streamline the process of adding new deviations' causal factors to the pipeline and improve the data quality. Overall, leveraging text describing the potential causes of historical manufacturing deviations at Amgen has been shown to be valuable to investigators and has the potential to improve how Amgen conducts deviation investigations.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Additional Figures



Executive Summary

Event Description
(Correction/Containment)
(Scoping Strategy)
(Product Impact)

Context/Background
Prior Occurrence Review

Root Cause Analysis
RCA Methodology
RCA Team Members

Causal Factor Table

5M Category	Potential CF	CF Confirmed?	Rationale
[Materials, Method, etc..]	Free text	[Yes, no, unconfirmed]	Free text

Discussion of Identified Root Cause(s)
Corrective Action (CA) and Preventative Action (PA) Discussion
(Effectiveness Verification Discussion)

Figure A-1: Overview of deviation report contents. Optional sections shown in parentheses.

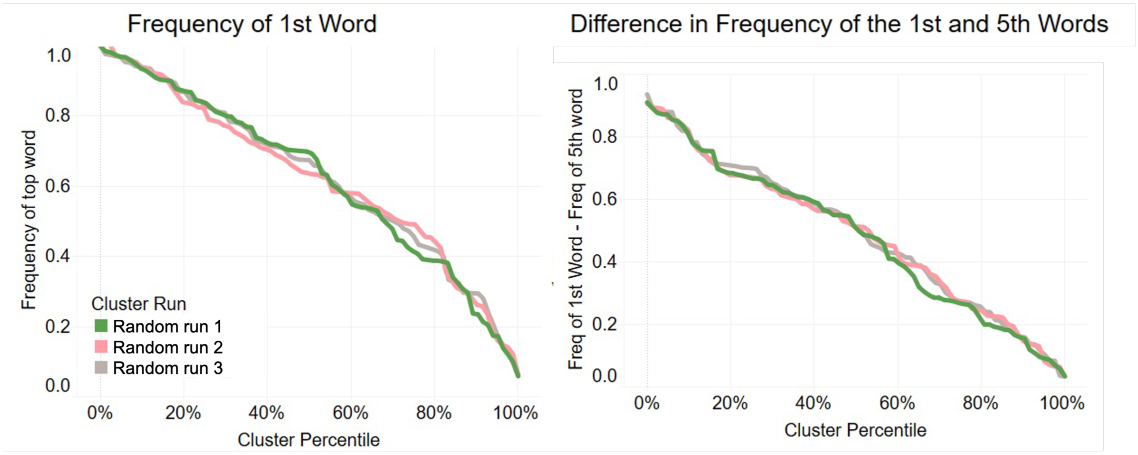


Figure A-2: Comparison of summary word frequency metrics for clustering runs using different random seeds but otherwise, identical parameters.

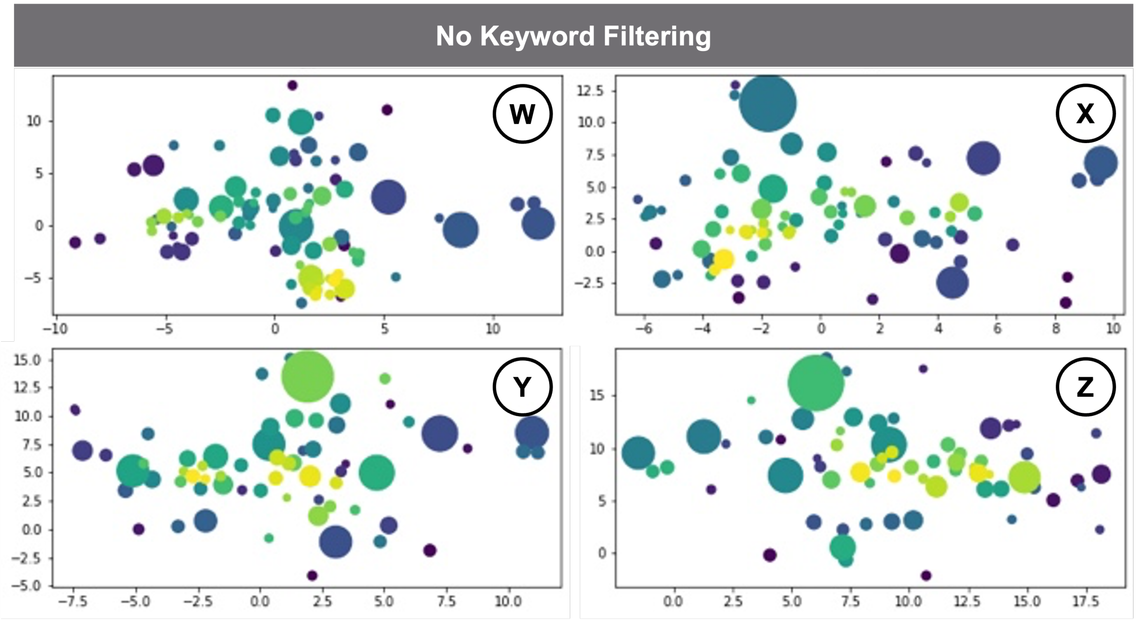


Figure A-3: Baseline distribution of causal factors in clusters for four different embedded text column options.

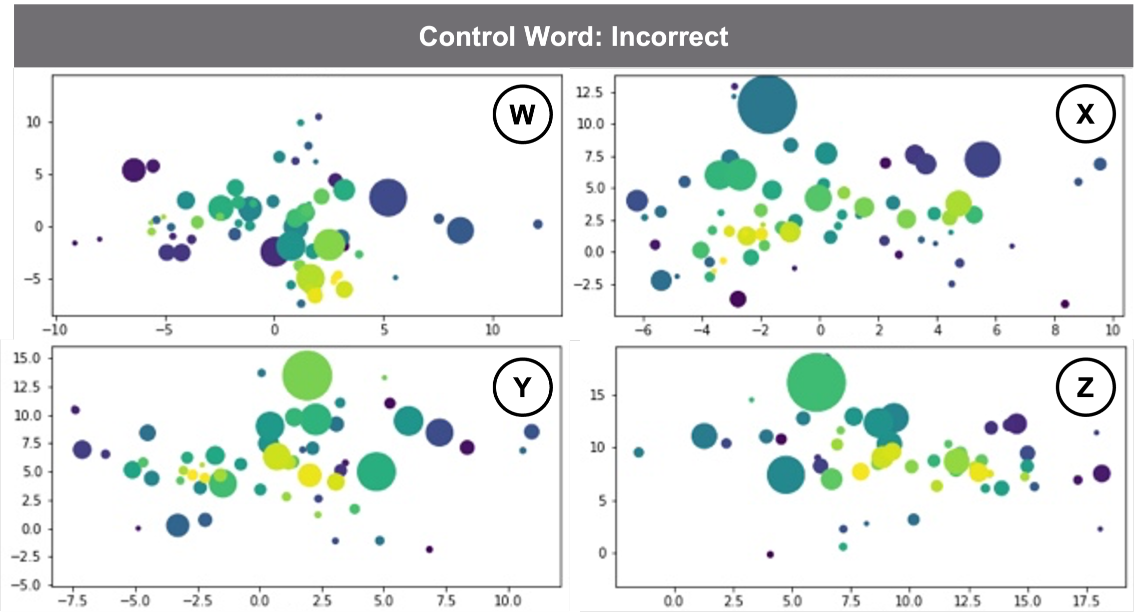


Figure A-4: Comparison of the distribution of the control word "incorrect" within causal factor clusters for four different embedded text column options. The word "incorrect" is not an important distinguishing keyword and therefore should appear generally across clusters rather than concentrated in a few. Here, we can see that for all columns, this condition is satisfied, indicating that none of the columns are generating clusters erroneously clustering around the word "incorrect".

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Francesca Alloatti, Luigi Di Caro, and Gianpiero Sportelli. "Real Life Application of a Question Answering System Using BERT Language Model". In: *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*. Stockholm, Sweden: Association for Computational Linguistics, Sept. 2019, pp. 250–253. DOI: `10.18653/v1/W19-5930`. URL: `https://aclanthology.org/W19-5930`.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3 (2003), pp. 993–1022.

[3] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172. ISBN: 978-3-642-37456-2.

[4] Ricardo Campos et al. "YAKE! Keyword extraction from single documents using multiple local features". In: *Information Sciences* 509 (2020), pp. 257–289. ISSN: 0020-0255. DOI: `https://doi.org/10.1016/j.ins.2019.09.013`. URL: `https://www.sciencedirect.com/science/article/pii/S0020025519308588`.

[5] K. R. Chowdhary. "Natural Language Processing". In: *Fundamentals of Artificial Intelligence*. New Delhi: Springer India, 2020, pp. 603–649. ISBN: 978-81-322-3972-7. DOI: `10.1007/978-81-322-3972-7_19`. URL: `https://doi.org/10.1007/978-81-322-3972-7_19`.

[6] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[7] Carlos A Escobar and Ruben Morales-Menendez. "Machine learning techniques for quality control in high conformance manufacturing environment". In: *Advances in Mechanical Engineering* 10.2 (2018).

[8] Stefan Feuerriegel and Nicolas Pröllochs. "Investor Reaction to Financial Disclosures across Topics: An Application of Latent Dirichlet Allocation". In: *Decision Sciences* 52.3 (Dec. 2018), pp. 608–628. DOI: `10.1111/deci.12346`. URL: `https://doi.org/10.1111/deci.12346`.

[9] Joanna Gallant. "Do You Make This Critical Root Cause Analysis (RCA) Mistake?" In: *Pharmaceutical Online* (Nov. 2016).

[10] Venkat N Gudivada, Dhana L Rao, and Amogh R Gudivada. "Information retrieval: concepts, models, and systems". In: *Handbook of statistics*. Vol. 38. Elsevier, 2018, pp. 331–401.

[11] Andre C Guerra and Jarka Glassey. "Machine learning in biopharmaceutical manufacturing". In: *European Pharmaceutical Review* 23.4 (Sept. 2018), pp. 62–65.

[12] Beakcheol Jang, Inhwan Kim, and Jong Wook Kim. "Word2vec convolutional neural networks for classification of news articles and tweets". In: *PLOS ONE* 14.8 (Aug. 2019). DOI: 10.1371/journal.pone.0220976. URL: https://doi.org/10.1371/journal.pone.0220976.

[13] A.O. Kirdar, K.D. Green, and A.S. Rathore. "Application of Multivariate Data Analysis for Identification and Successful Resolution of a Root Cause for a Bioprocessing Application". In: *Biotechnology Progress* 24.3 (June 2008), pp. 720–726. DOI: 10.1021/bp0704384. URL: https://doi.org/10.1021/bp0704384.

[14] Sebastian Kula, Michał Choraś, and Rafał Kozik. "Application of the BERT-Based Architecture in Fake News Detection". In: *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*. Ed. by Álvaro Herrero et al. Springer International Publishing, 2021, pp. 239–249. ISBN: 978-3-030-57805-3.

[15] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[16] Eric Langer. *5th annual report and survey of biopharmaceutical manufacturing capacity and production*. Bioplan Assn, 2008. ISBN: 1934106097.

[17] Fucun Li et al. "Ensemble machine learning systems for the estimation of steel quality control". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 2245–2252.

[18] Anna Lokrantz, Emil Gustavsson, and Mats Jirstrand. "Root cause analysis of failures and quality deviations in manufacturing using machine learning". In: *Procedia CIRP* 72 (2018). 51st CIRP Conference on Manufacturing Systems, pp. 1057–1062. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2018.03.229. URL: https://www.sciencedirect.com/science/article/pii/S2212827118303895.

[19] Qiuping Ma, Hongyan Li, and Anders Thorstenson. "A big data-driven root cause analysis system: Application of Machine Learning in quality problem solving". In: *Computers & Industrial Engineering* 160 (2021), p. 107580. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2021.107580. URL: https://www.sciencedirect.com/science/article/pii/S0360835221004848.

[20] Sue Marchant. "The self-investigating CAPA of the future". In: *Pharma Manufacturing* (June 2021).

[21] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861. URL: https://doi.org/10.21105/joss.00861.

[22] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[23] Basim Al-Najjar and Imad Alsyouf. "Improving effectiveness of manufacturing systems using total quality maintenance". In: *Integrated Manufacturing Systems* 11.4 (Jan. 2000), pp. 267–276. ISSN: 0957-6061. DOI: 10.1108/09576060010326393. URL: https://doi.org/10.1108/09576060010326393.

[24] Ricardo Silva Peres et al. "Multistage Quality Control Using Machine Learning in the Automotive Industry". In: *IEEE Access* 7 (2019), pp. 79908–79916. DOI: 10.1109/ACCESS.2019.2923405.

[25] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[26] Alberto Tellaeche and Ramón Arana. "Machine learning algorithms for quality control in plastic molding industry". In: *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. 2013, pp. 1–4.

[27] Bach Xuan Tran et al. "Modeling Research Topics for Artificial Intelligence Applications in Medicine: Latent Dirichlet Allocation Application Study". In: *J Med Internet Res* 21.11 (Nov. 2019). ISSN: 1438-8871. DOI: 10.2196/15511. URL: http://www.ncbi.nlm.nih.gov/pubmed/31682577.

[28] Jay B. Wish. "Biosimilars—Emerging Role in Nephrology". In: *Clinical Journal of the American Society of Nephrology* 14.9 (2019), pp. 1391–1398. ISSN: 1555-9041. DOI: 10.2215/CJN.01980218. URL: https://cjasn.asnjournals.org/content/14/9/1391.

[29] Lawrence X. Yu et al. "Understanding pharmaceutical quality by design". In: *The AAPS journal* 16.4 (July 2014), pp. 771–783. ISSN: 1550-7416. DOI: 10.1208/s12248-014-9598-3. URL: https://doi.org/10.1208/s12248-014-9598-3.

[30] Dongwen Zhang et al. "Chinese comments sentiment classification based on word2vec and SVMperf". In: *Expert Systems with Applications* 42.4 (2015), pp. 1857–1863. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.09.011. URL: https://www.sciencedirect.com/science/article/pii/S0957417414005508.