BENCHMARKING LARGE LANGUAGE MODELS FOR GXP HEALTHCARE COMPLIANCE

Vishweshwar Tyagi, Daoxing Zhang, Siqi He, Siwen Xie, Yihao Gao

{vt2353, dz2479, sh4190, sx2291, yg2820}@columbia.edu

ABSTRACT

In this project, we team up with J&J's Quality Assurance wing to benchmark large language models and assess their performance in classifying regulatory requirements from health authorities into J&J quality topics. We achieved an Exact Matching Ratio of 0.72 with BERT, 0.73 with Ada, 0.74 with Curie, and 0.753 with their ensemble on multi-label topic classification across 40 non-mutually exclusive quality topics. Finally, we also evaluate their text embeddings and notice a better cluster separation with fine-tuned models than their vanilla counterparts.

1. BACKGROUND AND AIM

The GxP regulatory environment is complex as different countries have regulations, and standardization is limited. These regulatory requirements must be manually parsed, analyzed, and classified to develop the J&J quality requirements. This is a time-consuming process where some automation is desired.

To tackle this problem, the Quality Assurance wing of J&J teamed up with OpenAI to build a fine-tuned GPT-3 model capable of binary classifying regulatory requirements as either *hygiene* or *other*. In this project, our goal is to scale the current implementation to a much larger dataset, incorporating 40 non-mutually exclusive quality topics, and move from a proof-of-concept into a minimal viable product, formulate metrics which align with business needs to evaluate text-embeddings from Large Language Models (LLM) like the GPT-3, use these metrics to benchmark GPT-3 against classical machine-learning methods as well as other LLM.

With the GxP regulatory documents, we will focus on two tasks - classification and clustering. Each row of the dataset is a regulation which can have multiple non-mutually exclusive labels and constitutes a multi-label classification problem. Unlike multi-class classification, where learning a single probability distribution over all classes is sufficient, we need to learn a different distribution for each class and evaluate it with metrics explicitly designed for this problem. We will leverage pre-trained LLM and fine-tune them to get the best results. To evaluate their embeddings, we will perform unsupervised clustering and compare them based on cluster-purity.

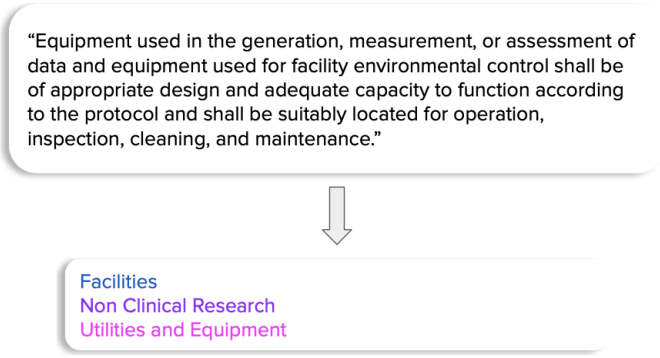

Fig. 1. Multi-label Classification

2. EXPLORATORY DATA ANALYSIS

Figure 1 demonstrates the classification task. Each row contains a regulation and its assigned labels. Each regulation can be either *Core, Non-core,* or *Obsolete* requirement.

Table 1 contains information about the dataset. For EDA, we drop the rows with missing labels, which aligns with the needs of the classification task. This leaves us with a dataset of $3,906$ rows and 13 columns. Since our project aims to classify regulations into different quality topics, we mainly explore samples with labels and their text. Data was processed for further analysis that included cleaning text and removing stopwords.

Fig 2 shows how many regulations are assigned multiple labels. Almost 60% of the regulations have exactly one label, while the maximum number of labels assigned to a regulation equals 14.

Total	**#**
Rows	10,232
Columns	13
Classes	48
Rows missing label	6,326

Table 1. Dataset

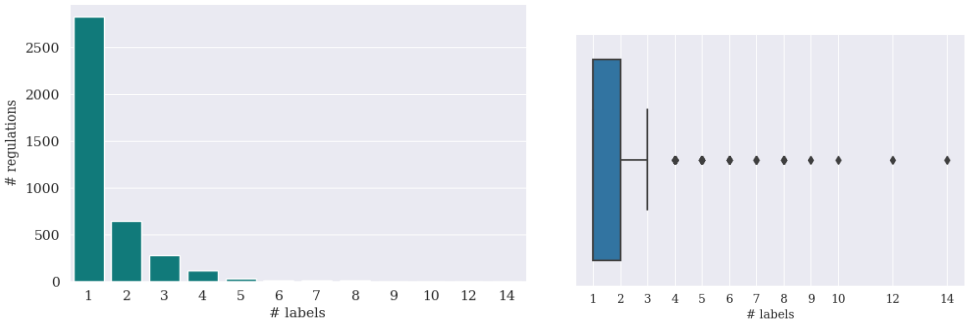

Fig. 2. How many regulations have more than one label?

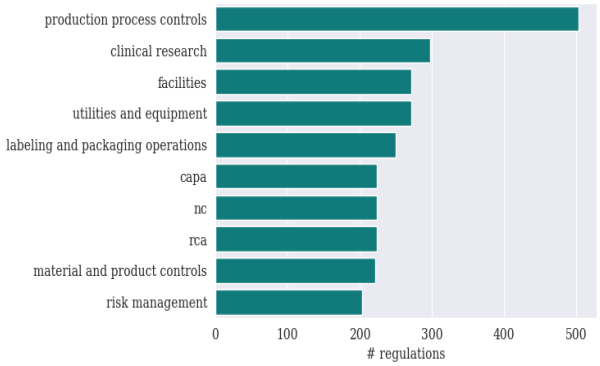

Fig. 3. Most frequent quality topics

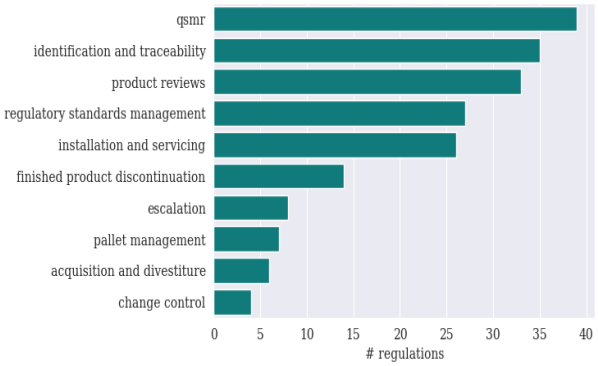

Fig. 4. Least frequent quality topics

Figure 3 and 4 respectively display the 10 most and least frequent quality topics. More than five hundred requirements are mapped to label *Production Process Controls*. Also over two hundred regulations are labeled *Clinical Research* and *Utilities and Equipment*, while less than ten requirements are categorized into label *Change Control*. It infers that the dataset is imbalanced based on the distribution of the number of regulations of each label. Moreover, we explore the relationship between regulation requirement type and label counts.

Figure 5 indicates the quality topics for regulations depend on requirement type. For core requirement, the top 3 most frequent labels are *Production Process Controls*, *Facilities*, and *Utilities and Equipment*; for type Non-core type, they are *Data Management*, *Risk Management*, and *Clinical Research*; and for type Obsolete, they are *Quality Management*, *CAPA(Corrective and Preventive Actions)* and *NC(non conformances)*.

After analyzing the target distribution, we perform text analysis. First, we conduct word frequency analysis. Figure 6 displays the most frequent n-grams. Top 5 common words found in texts are **quality**, **product**, **control**, **device** and **data**. 2-grams and 3-grams appear highly informative; word-TFIDF would be reasonable for building features.

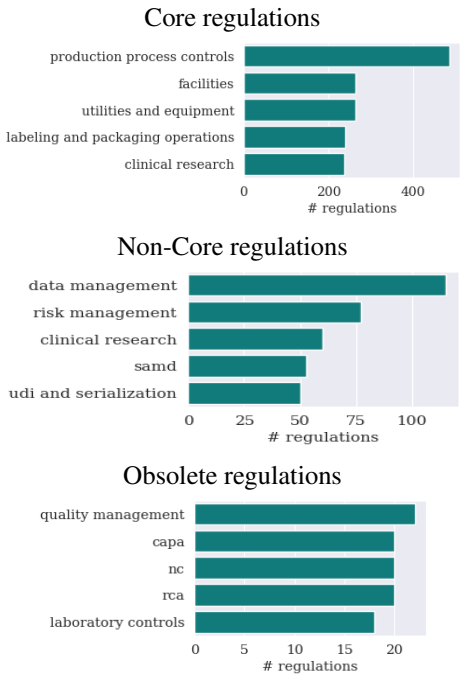

Fig. 5. Quality topics depend on requirement type

Fig. 6. Most frequent n-grams

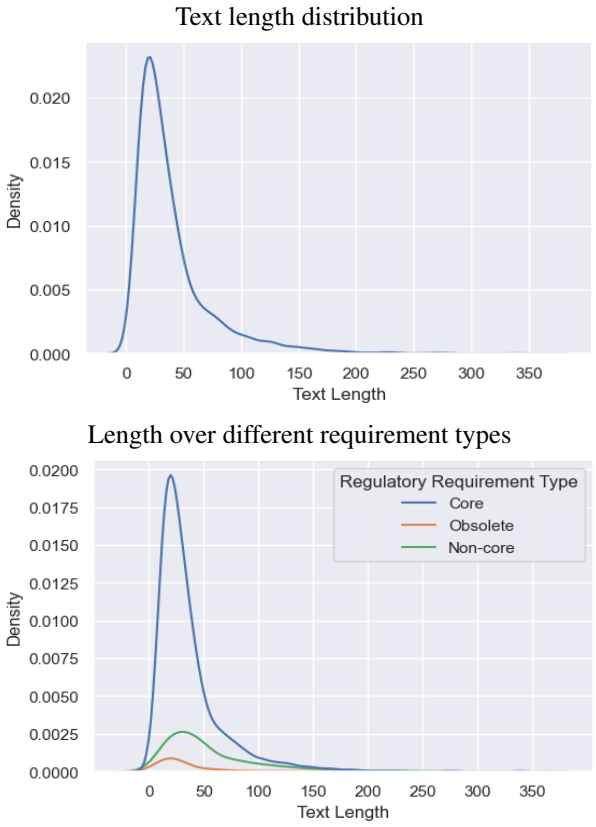

Fig. 7. Length of regulation requirements

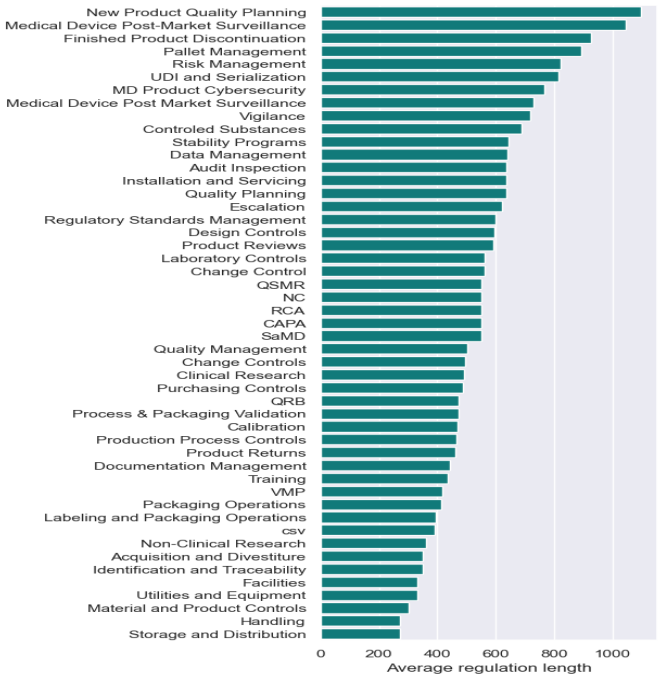

Fig. 8. Length of regulation requirements

Figure 7 suggests that the text length distribution is skewed right and centred around 40 to 50 words, which holds even irrespective of regulatory requirement type. This means that more regulations are shorter than average length and a few regulations are very lengthy.

According to Figure 8, the average length by label varies significantly: while documents related to *New Product Quality Planning* have over 1000 words, the average length of requirements labelled *Storage and Distribution* is less than 300 words. Below, we visualize n-grams for the top 3 most frequent labels.

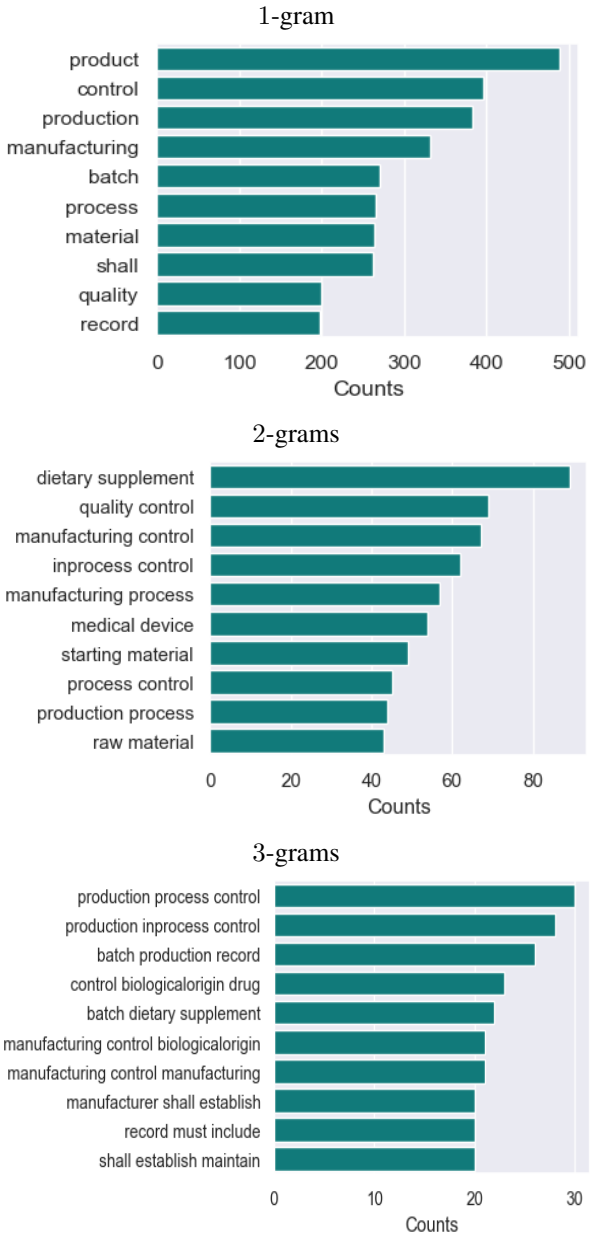

Fig. 9. Most frequent n-grams for Production Process Controls

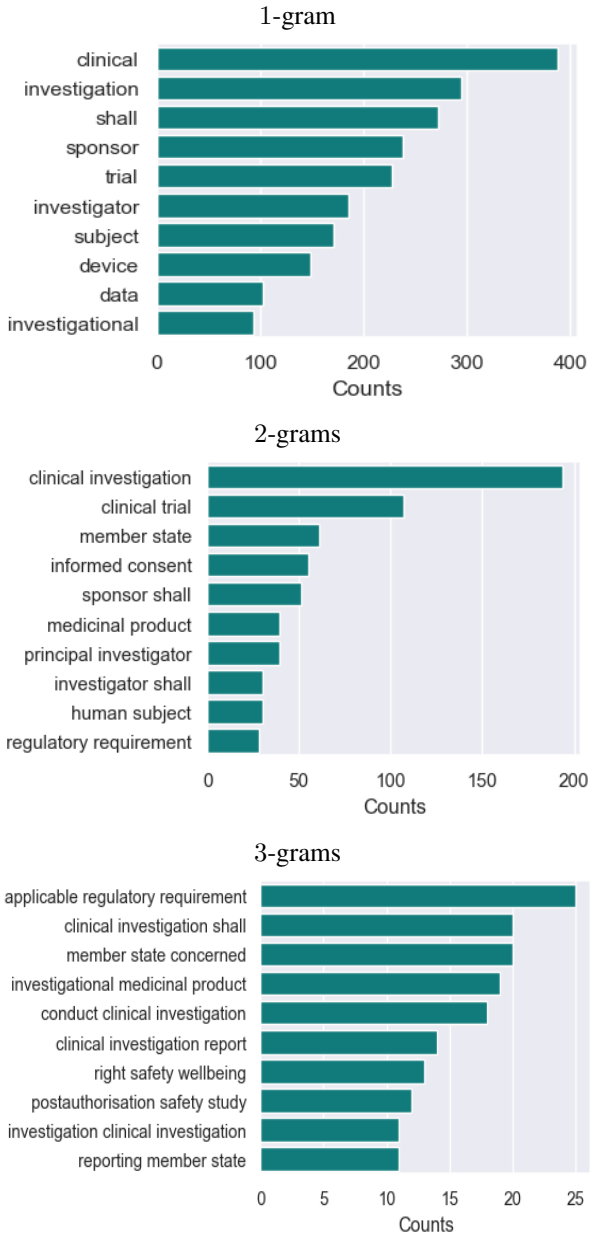

Fig. 10. Most frequent n-grams for Clinical Research

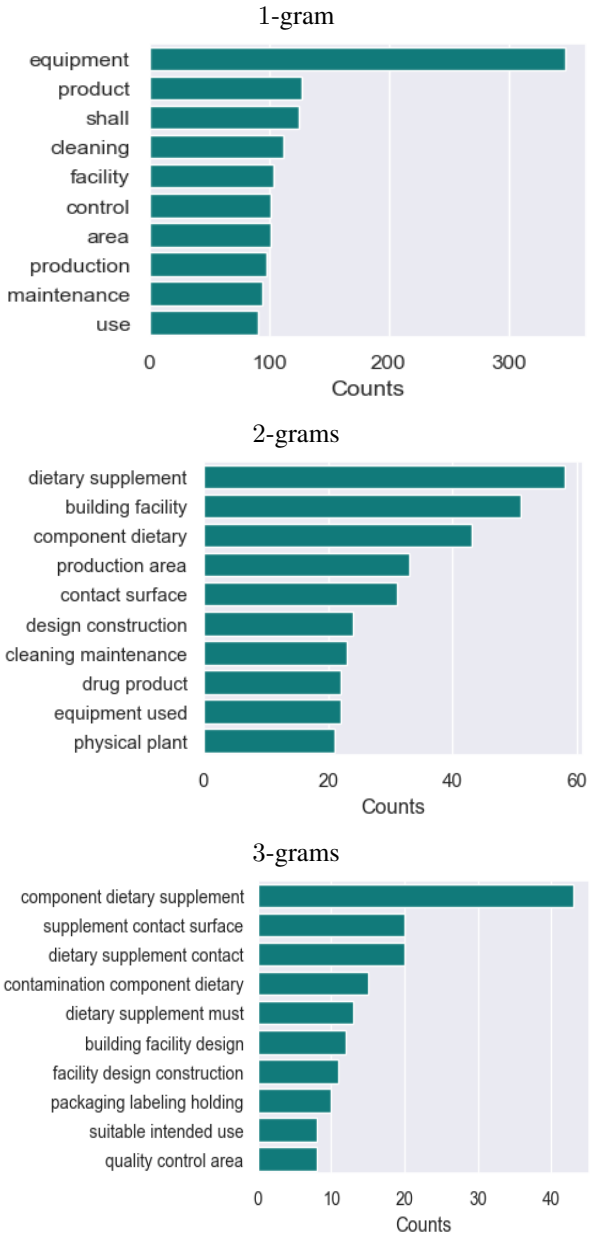

Fig. 11. Most frequent n-grams for Utilities and Equipment

From the n-gram analysis, we see that regulatory requirements are focused on **quality control**, **medical device**, **quality management** and so on. Looking through n-grams for requirements with the top 3 frequent labels, we find that even though some sequence of words are overlapping among the top 3 frequent labels, there are mostly distinct features: for *Production Process Controls*, its content is mainly about manufacturing; for *Clinical Research*, the text is more about the clinical and medical staff; and for *Utilities and Research*, the topic of documents are around equipment and product. Some overlap in words is expected since regulations are written in a structured manner and make use of specific words often.

Classification	**# Targets**	**Target cardinality**
Multiclass	1	> 2
Multilabel	> 1	2 (binary)
Multiclass-multioutput	> 1	> 2

Table 2. Classification Problems

We find that n-grams can be highly informative in assigning labels. This motivates us to build features using the word-TFIDF. We won't require character-TFIDF features since we don't see any spelling mistakes. Due to how regulations are structured, there are few n-grams that occur too often and are not informative. To avoid these, we will set a maximum frequency threshold on word-TFIDF. Additionally, we'll include 2 and 3-grams, as we saw these can help significantly.

3. METHODS

We discuss the methodology employed for multi-label classification and clustering regulations in order. We use metrics defined in section 4 to evaluate models and report their performance in section 5.

3.1. Classification

Recall from 1 the task of multi-label classification. Table 2 shows how the multi-label classification problem differs from the more common multi-class classification. It's defined as a classification task labeling each sample with m labels from n_{classes} possible classes, where $0 \leq m \leq n_{\text{classes}}$.

This can be thought of as predicting properties of a sample that are not mutually exclusive. Formally, a binary output is assigned to each class, for every sample. Positive classes are indicated with 1 and negative classes with 0. In our case, $n_{\text{classes}} = 40$, which we enumerate as T_j for $j \in \{1, 2 \dots n_{\text{classes}}\}$. The prediction task is formulated as:

Given text regulation R_j, assign an ordered vector,

$$\hat{L}_j = [\hat{L}_j^1, \hat{L}_j^2 \dots \hat{L}_j^{n_{\text{classes}}}]^T$$

such that $\hat{L}_j^i \in \{0, 1\}$, where $\hat{L}_j^i = 1$ if and only if the predictor believes quality topic T_i is applicable to regulation R_j

We divide the dataset into development (90%) and test set (10%). This section outlines the methods used for multi-label classification.

3.1.1. Baseline

For the baseline model, we first process the regulations. The steps followed in order are, remove HTML and punctuation, convert to lower case, remove URLs. We featurize the processed text using a word-tfidf vectorizer with 2 and 3

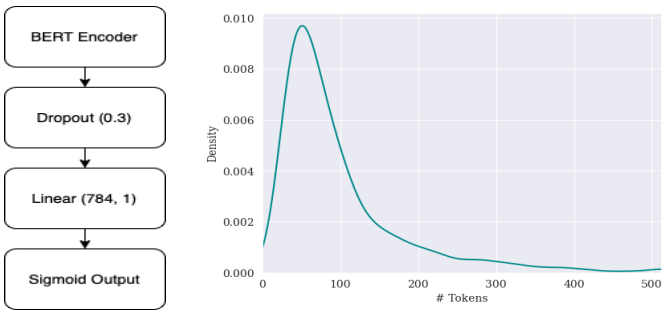

Fig. 12. BERT

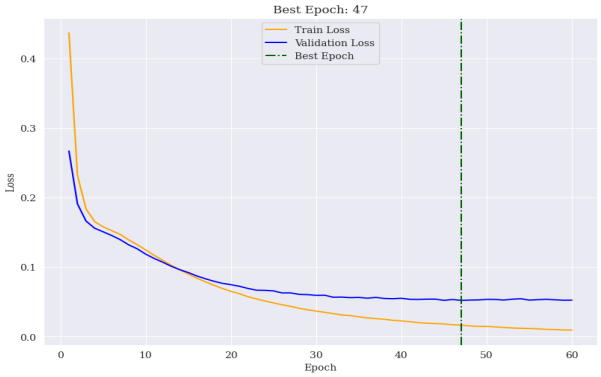

Fig. 13. BERT Train Validation Loss Curve

grams, minimum and maximum frequencies set to 0.005 and 0.725 respectively, and obtain features of 562 dimension on the development set. We train a random forest classifier on the development TF-IDF features and choose it as our baseline model.

3.1.2. BERT

BERT[1] is a transformer-based language model which has shown excellent performance across a wide range of natural language processing tasks, including classification.

Figure 12 displays a custom PyTorch model consisting of a classifier head placed on top of BERT. Table 3 lists the number of parameters.

We divide the development set into train (90%) and validation set (10%). We fine-tune the model on unprocessed regulations from the train set which are encoded by the BERT tokenizer. We set the maximum token count for the tokenizer based on Figure 12, listed in Table 3. We select the best epoch based on minimum validation loss, shown in Figure 13.

3.1.3. GPT-3

Fine-tuning GPT-3 requires prompt-completion dataset, displayed in Figure 14. Target labels are sorted alphabetically and mapped one-one to natural numbers. We use stop

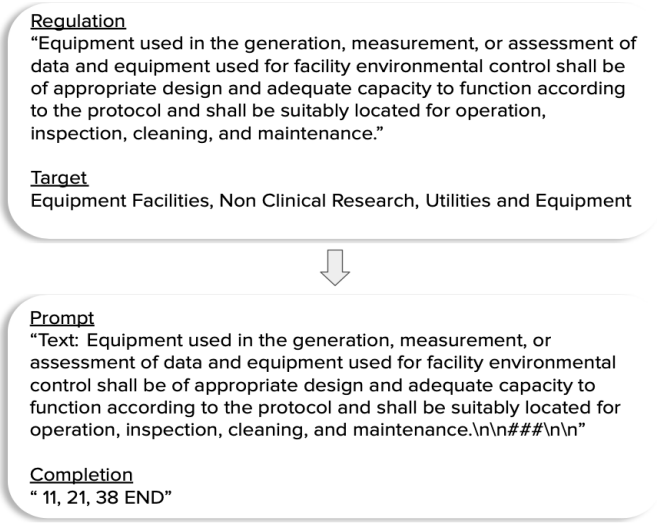

Fig. 14. GPT-3 Prompt and Completion

token *END* to prevent hallucination. We fine-tune Ada, Curie and Davinci from the GPT-3 series for 4 epochs along with validation.

3.2. Clustering

We evaluate embeddings of large language models on unsupervised task of clustering regulations, described in Figure 15. We use Elbow method, which minimizes distortion and inertia, to find the number of centroids for k-means. To visulize clustering results, we use TSNE and project embeddings to 2D space.

We extract fine-tuned embeddings of the BERT component of Figure 12 and compare their clustering performance against embeddings of vanilla BERT. To avoid data leakage, we perform clustering only on the test dataset. This poses a problem - for most quality topics, the number of regulations present in the test dataset are fairly low for any clustering algorithm to perform decently. To deal with this, we set a minimum threshold of 13 regulations, which reduces the number of quality topics to 8. Additionally, we get rid of regulations that map to more than one quality topic which allows us to

Parameter	**#**
Learning Rate	10^{-5}
Batch Size	32
Epochs	60
Maximum Token Count	256
Loss Criterion	Binary Cross Entropy
Optimizer	Adam

Table 3. BERT Hyperparameters

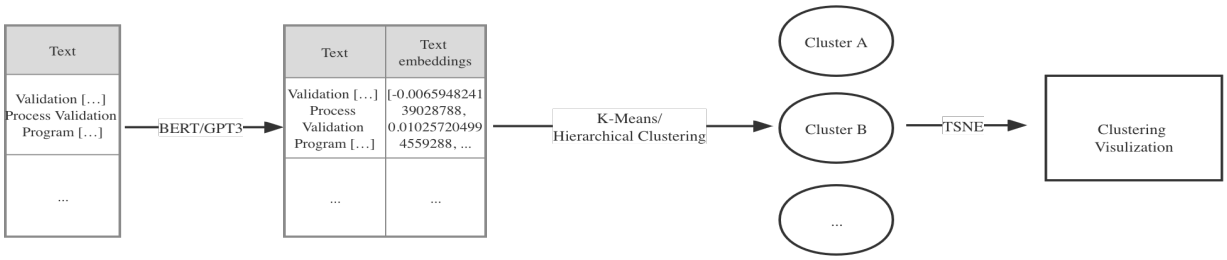

Fig. 15. Clustering

compare based on cluster-purity.

Unfortunately, OpenAI restricts extracting embeddings of fine-tuned models, but we can still compare embeddings of vanilla GPT-3 models against BERT's. We report the results in section 5.

4. EVALUATION

4.1. Classification

We enumerate the 40 quality topics alphabetically as $T_1, T_2 \dots T_{n_{\text{classes}}}$ with $n_{\text{classes}} = 40$. We have $N = 384$ regulations in the test set.

Given text regulation R_j, the ground truth is an ordered vector,

$$L_j = [L_j^1, L_j^2 \dots L_j^{n_{\text{classes}}}]^T$$

such that $L_j^i \in \{0, 1\}$, where $L_j^i = 1$ if and only if the quality topic T_i applies to regulation R_j.

We also define the set of true labels for regulation R_j, as

$$S_j = \{T_i \mid L_j^i = 1,\ i = 1, 2, \dots n_{\text{classes}}\}$$

and similarly, set of predicted labels as,

$$\hat{S}_j = \{T_i \mid \hat{L}_j^i = 1,\ i = 1, 2, \dots n_{\text{classes}}\}$$

To assess the performance of a predictor, we compare its prediction vector $\hat{L}_j$ defined in section 3.1 against the ground truth vector L_j and compute the following metrics:

- Hamming loss: $\frac{1}{N \times n_{\text{classes}}} \sum_{j=1}^{N} \sum_{i=1}^{n_{\text{classes}}} \textbf{xor}(L_j^i, \hat{L}_j^i) \in [0, 1]$

  computes the proportion of incorrectly predicted labels to the total number of labels. Here **xor** is the Exclusive-Or operator. Lower is better.

- Exact Matching Ratio / Accuracy: $\frac{1}{N} \sum_j \mathbf{1}_{[\hat{L}_j = L_j]} \in [0, 1]$

  This the most strict metric, indicating the percentage of samples that have all their labels classified correctly. The drawback is that it does not account for partially correct labels. Higher is better.

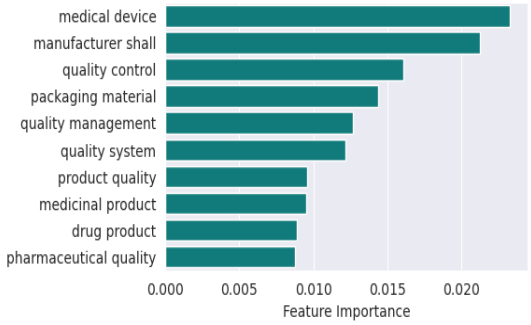

Fig. 16. Random Forest Important Features

- Precision: $\frac{1}{N} \sum_j \frac{|S_j \cap \hat{S}_j|}{|\hat{S}_j|} \in [0, 1]$, higher is better.
- Recall: $\frac{1}{N} \sum_j \frac{|S_j \cap \hat{S}_j|}{|S_j|} \in [0, 1]$, higher is better.
- F1 Score: $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \in [0, 1]$ is the harmonic mean of precision and recall. Higher is better.
- F2 Score: $\frac{5 \times \text{Precision} \times \text{Recall}}{(4 \times \text{Precision}) + \text{Recall}} \in [0, 1]$ is similar to $F1$ score but puts more emphasis on recall. Higher is better.

4.2. Clustering

Since we have access to ground truth, we can evaluate a clustering mechanism based on cluster purity. A cluster's purity is defined as the proportion of the most-frequent quality topic that belongs to it. The purity score is bounded above by 1, and higher values are desirable. Further, **every** quality topic should form the majority of **exactly one** cluster.

5. RESULTS

5.1. Classification

Table 4 displays the classification results. Figure 16 shows the important features uncovered by Random Forest. We notice LLM massively outperform Random Forest. Of all the LLM, Curie performs the best, which was a surprise

	Random Forest	BERT	Ada	Curie	Davinci	Ensemble
Hamming Loss	0.032	0.014	0.014	0.013	0.014	**0.012**
Exact Matching Ratio	0.359	0.706	0.730	0.740	0.727	**0.753**
Precision	0.511	0.851	0.863	0.875	0.855	**0.880**
Recall	0.551	**0.843**	0.828	0.839	0.837	0.839
F1 Score	0.506	0.827	0.830	0.842	0.832	**0.845**
F2 Score	0.525	0.831	0.825	0.837	0.831	**0.839**

Table 4. Benchmark Results

	BERT	Ada	Curie	Davinci
# Parameters	108M	350M	1.3B	175B
Embeddings Dimension	768	1024	4096	12288

Table 5. Large Language Models - Size Comparison

because Davinci, the biggest GPT-3 model from Table 5, is touted as the most powerful for most tasks. Perhaps, bigger is not always better.

We achieve the best results with an ensemble of BERT, Ada and Curie based on majority vote. Table 6 lists the label-wise benchmark results of the best-performing ensemble method.

5.2. Clustering

We discuss the results from section 3.2. Table 7 shows the purity scores on fine-tuned embeddings of BERT from k-means and hierarchical clustering. Every quality topic forms the majority of exactly one cluster with high purity scores. To set the number of centroids for k-means, we use the elbow method from Figure 17, which correctly identifies the number of centroids as the number of quality topics. Hierarchical clustering mechanism also identifies the number of clusters equal to the number of quality topics.

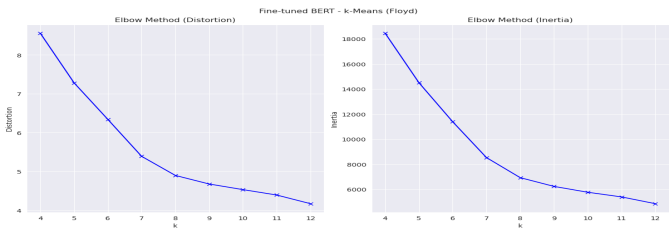

Fig. 17. Elbow Method - Fine-tuned BERT

We can also compare these to the embeddings results of vanilla BERT in Table 8, Ada in Table 9, Curie in Table 10, and Davinci in Table 11. Note that the $\times$ symbol means that the quality topic does not form a majority in any cluster and gets missed by the clustering mechanism. Additionally, multiple scores mean that the quality topic forms a majority in several clusters. Figure 18, 19, 20, and 21 show the elbow method for k-means on embeddings of vanilla BERT, Ada, Curie and Davinci respectively. We do not see a clear bend in any of these indicating that they all fail to correctly identify the number of centroids.

We see that fine-tuned BERT embeddings outperform vanilla BERT and GPT-3 embeddings and do not miss any topic or have duplicate topics. Using hierarchical clustering, it achieves the lowest 8.2% unclustered samples, with high purity scores. The results for GPT-3 models, on the other hand, needed to be more satisfactory. Some quality topics either did not surface as the majority labels or there were repeating topics clusters.

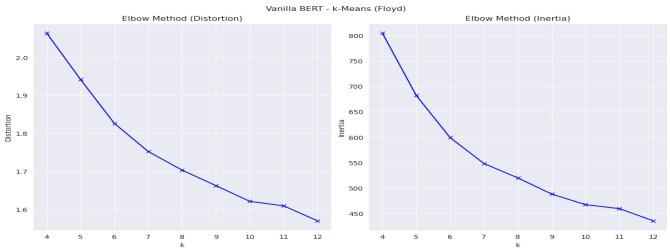

Fig. 18. Elbow Method - Vanilla BERT

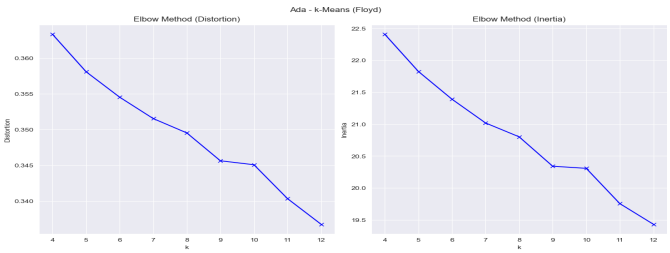

Fig. 19. Elbow Method - Ada Embedding

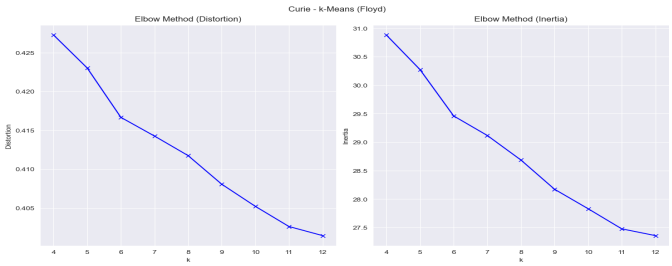

Fig. 20. Elbow Method - Curie Embedding

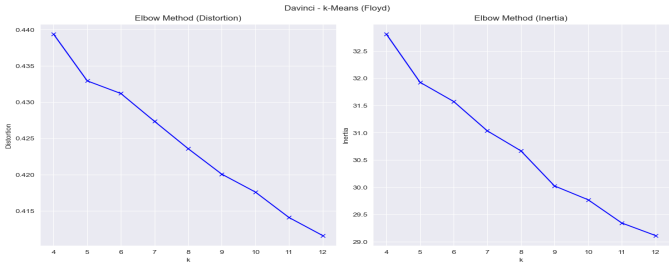

Fig. 21. Elbow Method - Davinci Embedding

6. DISCUSSION

6.1. Conclusion

We achieved the best results on the classification task with a majority-vote-based ensemble of BERT, Ada and Curie. Fine-tuning BERT helped achieve better cluster separation as compared to vanilla LLM.

6.2. Future Work

In the future, we could do multi-label classification using few-shot learning. We could not take it up this time because it cost us many credits. We could perform cluster analysis on GPT-3 fine-tuned embeddings if OpenAI were to make them accessible. We can also use the clusters to generate summaries and check them for coherence. We only used single-label samples for clustering, we could further scale this to include multi-label regulations and come up with metrics to evaluate the clusters when we have non-mutually-exclusive labels.

6.3. Ethical consideration

LLM are known to have bias, and their predictive prowess should be taken with a grain of salt. They should be tested and checked for bias before being deployed into production.

7. CONTRIBUTION

Please check Table 12

8. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

Quality Topic	Precision	Recall	F1 Score	F2 Score	Dev Samples	Test Samples
audit inspection	0.800	0.890	0.840	0.870	86	9
calibration	0.830	0.830	0.830	0.830	78	6
capa	0.830	0.750	0.790	0.770	202	20
change controls	0.880	0.780	0.820	0.800	71	9
clinical research	0.970	0.970	0.970	0.970	269	29
controled substances	1.000	0.830	0.910	0.860	40	6
csv	0.000	0.000	0.000	0.000	37	3
data management	0.760	1.000	0.870	0.940	150	13
design controls	1.000	0.910	0.950	0.930	132	11
documentation management	0.820	0.690	0.750	0.710	117	13
facilities	0.880	0.880	0.880	0.880	244	26
handling	0.780	0.580	0.670	0.610	99	12
identification and traceability	1.000	0.400	0.570	0.450	30	5
labeling and packaging operations	0.890	0.960	0.930	0.950	223	26
laboratory controls	0.730	0.790	0.760	0.770	143	14
material and product controls	0.760	0.830	0.790	0.810	195	23
md product cybersecurity	1.000	0.750	0.860	0.790	37	4
medical device post market surveillance	1.000	0.570	0.730	0.630	61	7
nc	0.830	0.750	0.790	0.770	202	20
new product quality planning	1.000	1.000	1.000	1.000	53	2
non clinical research	1.000	1.000	1.000	1.000	177	18
packaging operations	0.500	0.200	0.290	0.230	59	5
process & packaging validation	0.630	0.710	0.670	0.690	88	7
product returns	1.000	0.500	0.670	0.560	55	6
production process controls	0.980	0.690	0.810	0.730	441	61
purchasing controls	1.000	0.880	0.930	0.900	176	16
qrb	1.000	0.830	0.910	0.860	64	6
qsmr	1.000	0.400	0.570	0.450	33	5
quality management	0.810	0.850	0.830	0.840	173	20
quality planning	0.430	1.000	0.600	0.790	59	3
rca	0.830	0.750	0.790	0.770	202	20
risk management	0.780	0.610	0.680	0.640	180	23
samd	1.000	0.670	0.800	0.710	75	9
stability programs	0.830	0.910	0.870	0.890	113	11
storage and distribution	0.780	0.580	0.670	0.610	99	12
training	1.000	0.710	0.830	0.750	100	17
udi and serialization	1.000	0.600	0.750	0.650	80	10
utilities and equipment	0.880	0.880	0.880	0.880	244	26
vigilance	1.000	0.860	0.920	0.880	71	7
vmp	1.000	0.670	0.800	0.710	48	9

Table 6. Ensemble Label-wise Results

Quality Topic	K-Means (Floyd)	Hierarchical Clustering
Clinical Research	0.964	0.964
Data Management	1.000	1.000
Labeling and Packaging Operations	0.923	0.923
Material and Product Controls	0.933	0.929
Non Clinical Research	0.944	1.000
Production Process Controls	0.976	0.974
Quality management	0.923	0.900
Risk Management	0.786	1.000
Unclustered Samples	0 %	8.2 %

Table 7. BERT Embedding Results

Quality Topic	K-Means (Floyd)	Hierarchical Clustering
Clinical Research	0.350, 0.269, 0.214	0.350, 0.333
Data Management	0.375	×
Labeling and Packaging Operations	0.462	1.000
Material and Product Controls	×	0.500,0.400
Non Clinical Research	×	×
Production Process Controls	0.381, 0.324, 0.323	0.326
Quality management	×	×
Risk Management	×	×
Unclustered Samples	0 %	25.1 %

Table 8. Vanilla BERT Embedding Results

Quality Topic	K-Means (Floyd)	Hierarchical Clustering
Clinical Research	0.833, 0.588	1.000, 1.000
Data Management	0.706	1.000
Labeling and Packaging Operations	0.647	0.380
Material and Product Controls	×	×
Non Clinical Research	×	1.000
Production Process Controls	0.932, 0.714, 0.514	×
Quality management	0.588	×
Risk Management	×	×
Unclustered Samples	0 %	64.1 %

Table 9. Ada Embedding Results

Quality Topic	K-Means (Floyd)	Hierarchical Clustering
Clinical Research	0.812, 0.391	×
Data Management	0.417	×
Labeling and Packaging Operations	0.404	×
Material and Product Controls	×	×
Non Clinical Research	×	×
Production Process Controls	0.615,0.471	0.257
Quality management	×	×
Risk Management	1.000	0.750
Unclustered Samples	0 %	34.7 %

Table 10. Curie Embedding Results

Quality Topic	K-Means (Floyd)	Hierarchical Clustering
Clinical Research	0.923, 0.214	1.000, 1.000
Data Management	1.000	×
Labeling and Packaging Operations	×	×
Material and Product Controls	×	×
Non Clinical Research	0.484	×
Production Process Controls	0.542, 0.451, 0.385	×
Quality management	0.500	×
Risk Management	×	×
Unclustered Samples	0 %	41.3 %

Table 11. Davinci Embedding Results

Group Member	Contributions
Vishweshwar Tyagi	Data exploration, baseline modeling, Testing BERT/Ada/Curie/Davinci on classification, Evaluating embeddings, Poster Session PR Reviews, handling merge conflicts
Yihao Gao	Paper reviews, update project details on github, Presentation to J&J, exploratory data analysis PR review, GPT-3 few-shot learning and fine-tuning GPT-3 and BERT embedding, Poster session, final reports
Siqi He	Paper reviews, exploratory data analysis, Presentation to J&J, PR reviews GPT-3 few-shots learning and fine-tuning GPT-3 embedding, Poster session Write final reports
Siwen Xie	Paper reviews, exploratory data visualization, Presentation to J&J, PR reviews BERT model fine-tuning, Legal BERT Embedding Final Report, Poster session
Daoxing Zhang	Paper reviews, data exploration, Presentation to J&J, PR reviews Write final report, Embedding, Poster session GPT-3 few-shots learning and fine-tuning

Table 12. Contribution