Guidelines Matching

Qinglin Chen[*]
University of Bern
Bern, Switzerland
qinglin.chen

Hamza Yazbeck[*]
University of Bern
Bern, Switzerland
Hamza.Yazbeck

Chenrui Fan[*]
University of Bern
Bern, Switzerland
chenrui.fan

Abstract

Matching requirements with sustainability guidelines is a crucial task in automating regulatory compliance verification. In this study, we present a system for matching requirements with sustainability guidelines using pre-trained embeddings and multiple retrieval pipelines. Our approach begins with GPT embeddings for cosine similarity matching and extends to advanced re-ranking strategies, including cross-encoder and GPT-based methods with Chain of Thought (CoT) reasoning and Few-shot prompting. To address the scarcity of ground truth (GT) data and the limitations of traditional metrics, we propose an F-test-based metric, transforming the task into a binary classification problem for robust evaluation. By balancing Recall@K and the F-test, our method achieves satisfactory performance, highlighting the effectiveness of GPT-based re-ranking strategies in low-resource compliance verification tasks. The code for This project is available at `https://github.com/Sosekie/GuidelineMatching`.

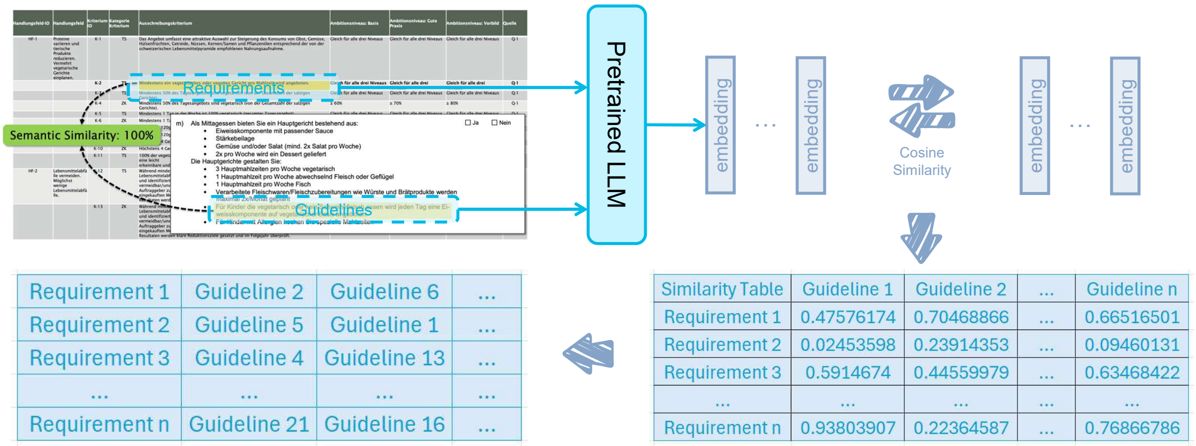
1 Introduction

Ensuring compliance with sustainability guidelines has become an increasingly critical task for organizations striving to meet regulatory and ethical standards. This challenge involves matching text fragments from documents, such as Call for Tenders (CFT), with relevant sustainability guidelines, a process that is essential for automating compliance verification in text-intensive environments. Given the manual effort required to handle large volumes of text, an automated system that can accurately identify matching guidelines is of great practical importance.

The primary goal of this project is to develop and evaluate a system capable of identifying relevant sustainability guidelines for specific text fragments in annotated documents. Initially, the system focuses on identifying guideline matches without assessing fulfillment, with the possibility of extending the scope to evaluate compliance. This task leverages pre-trained machine learning models and advanced retrieval techniques, reflecting the experimental nature of the problem and the need for comprehensive literature research.

To support this effort, we draw insights from prior work, including regulatory compliance verification using large language models [1], compliance checking of GDPR agreements [2], and natural language inference (NLI) methods for matching privacy policies with regulations [3]. These studies provide valuable frameworks for building and evaluating automated compliance systems on annotated datasets.

[*] Equal contribution.



Requirement 1	Guideline 2	Guideline 6	...
Requirement 2	Guideline 5	Guideline 1	...
Requirement 3	Guideline 4	Guideline 13	...
...	...	...	...
Requirement n	Guideline 21	Guideline 16	...

Similarity Table	Guideline 1	Guideline 2	...	Guideline n
Requirement 1	0.47576174	0.70468866	...	0.66516501
Requirement 2	0.02453598	0.23914353	...	0.09460131
Requirement 3	0.5914674	0.44559979	...	0.63468422
...	...	...	...	...
Requirement n	0.93803907	0.22364587	...	0.76866786

Figure 1.1: Structure.

Our proposed method adopts a straightforward and effective approach to guideline matching. By utilizing pre-trained embeddings, we avoid the need for fine-tuning while maintaining strong performance. To address the scarcity of ground truth (GT) data, we introduce a novel evaluation metric that combines Recall@K with the F-test, effectively analyzing the system's ability to distinguish between relevant and irrelevant matches. This enables robust evaluation in low-resource scenarios. In summary, our contributions are as follows:

- Leverage existing pre-trained models, eliminating the need for task-specific fine-tuning.
- Propose a combined metric using Recall@K and the F-test to handle scenarios with minimal ground truth annotations, providing a robust analysis of model performance on relevant and irrelevant data.
- Present a simple yet generalizable solution to the guidelines matching problem, applicable across diverse use cases.

2 Methodology

The primary objective of this work is to develop a robust system for matching requirements with sustainability guidelines. As illustrated in Figure 1.1, our approach leverages a shared pre-trained language model (LLM) to encode both requirements and guidelines into a common embedding space. This embedding space allows for efficient computation of semantic similarity using methods such as cosine similarity, facilitating accurate retrieval of relevant guidelines for a given requirement.

2.1 Core Idea

Our methodology begins by encoding both requirements and guidelines using the same pre-trained language model. This ensures that both types of textual data are represented in a semantically meaningful and consistent embedding space. Once the embeddings are generated, we compute the pairwise cosine similarity between requirements and guidelines to rank the guidelines based on their semantic relevance to each requirement. The ranked similarity scores form the foundation of our retrieval system, which is further refined through additional re-ranking strategies in more advanced pipelines.

2.2 Proposed Pipelines

To systematically evaluate the effectiveness of different retrieval strategies, we experiment with the following pipelines:

- **Baseline: GPT as a Matching Oracle.** In this approach, we directly utilize GPT to match each requirement with potential guidelines by prompting the model with the requirement text and asking it to identify the matching guidelines. This serves as a baseline to compare against embedding-based methods.

- **Pipeline 1: Cosine Similarity.** Using a pre-trained LLM, we generate embeddings for both requirements and guidelines. We then compute cosine similarity between each requirement and all guidelines, ranking the guidelines based on their similarity scores. This pipeline provides a straightforward and computationally efficient retrieval method.

- **Pipeline 2: Cosine Similarity with Cross-Encoder Re-ranking.** Building on the previous pipeline, we first rank guidelines using cosine similarity. For the top-k candidates (e.g., top-10), we employ a cross-encoder architecture using the pre-trained LLM to refine the ranking. The cross-encoder jointly encodes a requirement and a guideline, allowing for more context-aware scoring of semantic relevance.

- **Pipeline 3: Cosine Similarity with GPT Re-ranking.** Similar to Pipeline 2, this approach uses cosine similarity to pre-rank guidelines. However, instead of a cross-encoder, we utilize GPT for re-ranking by providing the top-k guidelines as input and asking the model to determine the most relevant guideline(s) for each requirement.

- **Pipeline 4: Cosine Similarity with Chain of Thought GPT Re-ranking.** This pipeline enhances Pipeline 3 by incorporating Chain of Thought (CoT) reasoning. Here, GPT is prompted to explain its reasoning step-by-step for each candidate guideline, aiming to improve the interpretability and accuracy of the re-ranking process.

- **Pipeline 5: Cosine Similarity with Chain of Thought and Few-shot GPT Re-ranking.** Extending Pipeline 4, this approach includes few-shot examples during GPT prompting to further guide the model in its reasoning and decision-making process. The examples illustrate how requirements and guidelines are matched, enhancing GPT's ability to re-rank candidates effectively.

2.3 Embedding and Retrieval Process

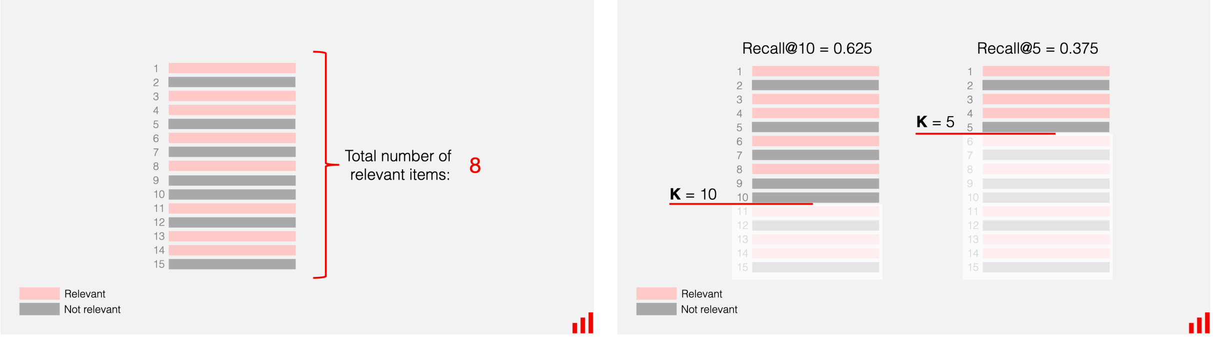
For all pipelines involving embeddings, we use pre-trained LLMs without fine-tuning to generate fixed-length vector representations for requirements and guidelines. The cosine similarity between these vectors serves as the initial ranking mechanism. This embedding-based approach provides a simple yet effective baseline, which can be further improved through re-ranking techniques as described above.

3 Experiment

3.1 Setup

EXPERIMENTAL ENVIRONMENT All experiments were conducted on a single workstation equipped with an Intel Core i7-13700KF processor and an NVIDIA GeForce RTX 4090 GPU. The software environment included Python 3.10, CUDA 11.8, and the corresponding PyTorch framework, ensuring efficient computation and hardware-accelerated operations for both the large language model and latent diffusion model tasks.

EXPANDING THE DATASET WITH ADDITIONAL UNRELATED INSTANCES In order to more thoroughly evaluate the model's ability to recognize irrelevant sentences, we supplemented our original dataset with an additional 30 sentences that are unrelated to "sustainability." This approach ensures a more balanced distribution of positive and negative samples.



(a) A list of the top 10 recommendations and a total of 8 items in the dataset are actually relevant.

(b) Zoom in on the first 5 suggestions. In this shorter list, we have only 3 relevant recommendations.

Figure 2.1: If the system includes 5 relevant items within the top 10 recommendations, the recall at rank 10 is 62.5%, capturing 5 out of the 8 relevant items in the dataset. Focusing on the top 5 recommendations, the system captures only 3 relevant items. This results in a recall at rank 5 of 37.5%, indicating that the system retrieves less than half of the relevant items in this shorter list.

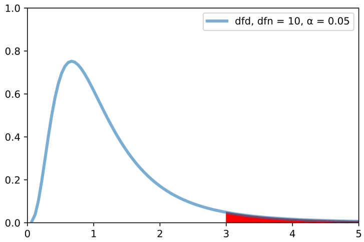


Figure 3.1: F-test distribution.

Repeated GPT experiments with averaged results Given that GPT outputs may vary between runs, we conducted multiple iterations of the same experiment and then averaged the results, thereby mitigating the potential impact of randomness on our performance estimates.

3.2 Evaluation Metrics

To evaluate our model's performance comprehensively, we employ two distinct metrics tailored to the dataset's characteristics: Recall@K for requirements with ground truth (GT) and an F-test for requirements without GT. These metrics effectively assess retrieval accuracy and the model's ability to handle edge cases.

Recall@K for Requirements with GT For requirements that have at least one corresponding guideline (i.e., with GT), we use Recall@K to measure retrieval performance, which is shown in Figure 2.1. Recall@K quantifies the proportion of relevant guidelines successfully retrieved within the top K ranked results. The formula is defined as follows:

$$\text{Recall@K} = \frac{\text{Number of relevant items in the top } K}{\text{Total number of relevant items}}$$

This metric provides insight into the system's effectiveness in prioritizing relevant guidelines among its top recommendations.

F-test for Requirements without GT For requirements that lack any corresponding guidelines (i.e., without GT), we employ an F-test to evaluate the model's ability to identify such cases, which is shown in Figure 3.1. The F-test tests the null hypothesis that no guidelines are similar to the given requirement. The evaluation involves the following steps:

Table 3.1: Performance Comparison of Baseline and Pipelines. Metrics include Recall@10, F Accuracy (Unrelated Acc), and Running Time (seconds).

Method	**Recall@10**	**F Accuracy**	**Running Time (seconds)**
Baseline: GPT as a Matching Oracle	0.046±0.023	0.511±0.021	1570.129±1096.301
Pipeline 1: Cosine Similarity	**0.527±0.000**	0.550±0.000	**1.738±0.383**
Pipeline 2: + Cross-Encoder Re-ranking	0.083±0.000	0.383±0.000	5.707±0.513
Pipeline 3: + GPT Re-ranking	0.340±0.069	**0.572±0.016**	272.409±0.575
Pipeline 4: + Chain of Thought	0.202±0.153	0.567±0.036	542.385±0.357
Pipeline 5: + Few-shot GPT Re-ranking	0.098±0.060	0.539±0.028	2066.009±15.907

- Grouping requirements into two categories: those with GT and those without GT.
- Calculating within-group variances (similarity scores among requirements in the same group) and between-group variances (similarity scores across the two groups).
- Computing a p-value based on the variance ratios. If the p-value exceeds 0.05, the null hypothesis is accepted, indicating no significant similarity between the requirement and any guideline. Otherwise, the null hypothesis is rejected, suggesting that at least one guideline is similar to the requirement.

By leveraging prior knowledge of which requirements have GT, this process is framed as a binary classification problem. The classification accuracy evaluates the model's ability to identify requirements with no matching guidelines.

3.3 Quantitative Analysis

Table 3.1 reports the experimental results for the baseline and our proposed pipelines, showcasing their performance in terms of Recall@10, F Accuracy, and running time. Overall, the pipelines demonstrate varying strengths:

- **Pipeline 1** achieves the highest Recall@10 (0.527±0.000) among all methods while also exhibiting the shortest running time (1.738±0.383 seconds). These findings suggest that a simple cosine similarity–based retrieval can be highly efficient and effective at retrieving the majority of relevant guidelines, despite its relatively basic approach.
- **Pipeline 3** attains the highest F Accuracy (0.572±0.016). Incorporating GPT for re-ranking enhances the model's ability to distinguish between requirements with and without matching guidelines. This improvement highlights GPT's strengths in understanding semantic nuances, although it comes with a higher computational cost (272.409±0.575 seconds).
- **Pipeline 5** requires the most extensive running time (2066.009±15.907 seconds), largely due to few-shot GPT re-ranking in combination with Chain of Thought prompts. Despite the additional complexity, its performance in Recall@10 (0.098±0.060) and F Accuracy (0.539±0.028) does not surpass the best performers in these categories. This suggests that while few-shot prompting can provide richer contextual understanding, it may also introduce additional overhead that does not always translate into improved retrieval performance.
- Notably, the **baseline** approach using GPT as a matching oracle (0.046±0.023 Recall@10, 0.511±0.021 F Accuracy) exhibits a comparatively large variance in running time (1570.129±1096.301 seconds), implying sensitivity to prompt variability and the model's inherent randomness.

3.4 Qualitative Analysis

In addition to the quantitative metrics, we also examined sample outputs to gain deeper insights into how our system retrieves and ranks guidelines. Below is an illustrative example, demonstrating how a requirement related to promoting the consumption of fruits, vegetables, and plant-based products is matched with potential guidelines:

> *Requirement: "TS Das Angebot umfasst eine attraktive Auswahl zur Steigerung des Konsums von Obst, Gemüse, Hülsenfrüchten, Getreide, Nüssen, Kernen/Samen und Pflanzenölen entsprechend der von der schweizerischen Lebensmittelpyramide empfohlenen Nahrungsaufnahme."*
>
> *Guideline 1: "To match the guideline regarding the attractive selection of food items to increase the consumption of fruits, vegetables, legumes, grains, nuts, seeds, and plant oils according to the Swiss food pyramid, we can identify the most relevant requirements from the provided list. Here's a ranking based on relevance:"*
>
> *Guideline 2: "Die Qualität der Ernährung an den Bieler Tagesschulen und Kitas neu entsprechend dem Reglement «Gesunde Ernährung» sichergestellt werden."*
>
> *Guideline 3: "This requirement directly relates to ensuring the quality of nutrition, which aligns with the guideline's focus on increasing the consumption of recommended food groups."*
>
> *Guideline 4: "Sie verwenden Öle, Gewürze, Streusalze, Zucker, Backwaren, wie auch weitere Basisprodukte und Nahrungsmittel für die Zwischenmahlzeiten zu 50% (an Kosten gemessen) in der Qualität «Bio» und (bei Produkten aus der 3. Welt) mit dem Gütesiegel «Max Havelaar»."*

The *Requirement* emphasizes increasing the consumption of specific food groups following the Swiss food pyramid. The *Guidelines 1 to 4* represent the retrieved statements or explanations linked to this requirement, each providing a different angle of relevance. Specifically:

- **Guideline 1** references the Swiss food pyramid and sets the context for ranking related requirements.
- **Guideline 2** focuses on ensuring a certain standard of nutrition in schools, aligning with the promotion of healthy eating.
- **Guideline 3** explicitly emphasizes increasing recommended food groups, thus reinforcing the nutritional goal.
- **Guideline 4** addresses the use of oils and various staple food items with an emphasis on sustainable sourcing (Bio, Max Havelaar), indicating an alignment between healthy eating and ecological considerations.

From these examples, we observe that the ranking process successfully retrieves guidelines related to improving nutritional quality, adhering to specific dietary recommendations (e.g., the Swiss food pyramid), and incorporating sustainability practices.

4 Conclusion

In this paper, we explored several embedding-based pipelines for matching requirements to sustainability guidelines. Our results indicate that a simple cosine similarity baseline can already achieve acceptable performance with minimal computation, whereas more advanced re-ranking strategies (including cross-encoders and GPT with Chain of Thought and Few-shot prompting) provide nuanced improvements at higher computational cost. Overall, embedding-based retrieval remains a strong foundation for guideline matching, but better domain adaptation, fine-tuning on larger, domain-specific datasets, and further refinement of interpretability (e.g., explainable re-ranking) are promising future directions. This line of work can ultimately facilitate more robust and transparent automation of sustainability compliance.

References

[1] Armin Berger, Lars Hillebrand, David Leonhard, Tobias Deußer, Thiago Bell Felix De Oliveira, Tim Dilmaghani, Mohamed Khaled, Bernd Kliem, Rudiger Loitz, Christian Bauckhage, et al. Towards automated regulatory compliance verification in financial auditing with large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4626–4635. IEEE, 2023.

[2] Orlando Amaral, Sallam Abualhaija, and Lionel Briand. Ml-based compliance verification of data processing agreements against gdpr. In *2023 IEEE 31st international requirements engineering conference (RE)*, pages 53–64. IEEE, 2023.

[3] Amin Rabinia and Zane Nygaard. Compliance checking with nli: Privacy policies vs. regulations. *arXiv preprint arXiv:2204.01845*, 2022.