LLMs in Health Sciences

Sai Satwik Dikkala and **Venkata Sai Vikas Katuru**
The State University of New York at Buffalo
saisatwi@buffalo.edu, vkaturu@buffalo.edu

Abstract

In this paper, we try to support the Large Language Models (LLMs) to accurately assess statements within Clinical Trial Reports (CTRs), focusing on the breast cancer trials. Our model task is a subtask SemEval-2024 Task 2 that targets critical features of these reports, such as trial outcomes, adverse effects, eligibility criteria, and treatment specifics, so that it enhances its capability in analyzing, interpreting, and drawing precise conclusions from the content of CTRs. The main task is to classify each pair of a clinical trial report as a text classification task and its related statement, determining if the statement is an "Entailment" or "Contradiction" to the information presented in the CTR. Our project's core objective is to develop a model based on LLMs capable of automating the assessment of CTRs. This model aims to improve the accuracy and efficacy of analyzing breast cancer CTRs, thereby supporting medical professionals in their research and treatment planning. By automating this process, we will be facilitating a more informed approach to evaluating the vast clinical trial reports.

1 Introduction

In the current scenario, in the field of oncology, clinical trial reports (CTRs) are the pillars for advancing the patient's care, mainly in the case of breast cancer treatment. CTRs provide complete and exhaustive details on eligibility criteria, trial methodologies, treatment outcomes, adverse effects, etc., offering thorough knowledge important for guiding clinical decisions. However, the complexity of the CTRs poses a significant challenge for healthcare clinicians, requiring a significant amount of time and knowledge for in-depth analysis. The complex information contained in the CTRs makes this process prone to human error. This project proposes an approach that simplifies the analysis of breast cancer CTRs by utilizing the advantages of natural language processing and artificial intelligence in order to overcome the abovementioned issues. Our aim is to develop a system that can independently extract and analyze information from the CTRs with improved accuracy and efficiency. This reduces the workload on medical clinicians by significantly increasing accuracy and speeding up the decision-making process in patients's care. Two main complimentary computational strategies—fine-tuning of pre-trained language models and utilizing zero-shot learning techniques—are core to our methodology. In the former case, existing large language models (LLMs) are customized to the specific domain of breast cancer CTRs, improving their ability to appropriately identify and interpret relevant information. The latter, zero-shot learning, presents a feasible path toward the model's ability to make informed assessments regarding previously unseen data, a skill in the dynamic field of clinical research. The dataset taken is a large dataset of breast cancer CTRs that is taken from the reputable database Clinical Trials Dataset. We have added expert annotations to these CTRs that classify the logical relationships between reports and the corresponding statements as either "Entailment" or "Contradiction". This detailed analysis is crucial for evaluating the consistency and faithfulness of CTRs, which serve as the basis for our model's testing and validation processes. The primary objective of this research is to establish an approach to analyzing breast cancer CTRs and improve the efficiency, accuracy, and accessibility of these crucial records for medical professionals. By achieving this, we hope to contribute to the larger objectives of precision medicine and evidence-based practice, as well as have a significant positive impact on the quality of patient care in breast cancer treatment. This work represents a major advancement in the use of NLP and AI technologies in the medical field, with the goals of SemEval-2024 Task 2.

2 Related Work

The Natural Language Inference (NLI) task is part of language understanding. The task is to figure out if one sentence is logically following another and determine whether there is 'entailment' or 'contradiction' between the premise and hypothesis. In the earlier days, people used rule-based approaches for these tasks. Then, large language models came into play and have shown promising results even without domain-specific training, especially in medical domain tasks.

Nowadays, plenty of resources are available for clinical NLI. Despite having enough data, LLMs faced challenges when dealing with quantitative data and the numerical operations of NLI tasks. This happens when we use LLMs trained on general topics for medical-related tasks. To overcome these challenges, researchers have started performing different techniques like zero-shot setting, few-shot setting, and fine-tuning. Among these techniques, fine-tuning is one of the most promising approaches, where the model is pre-trained on domain-specific data so that it becomes more adept at understanding the intricate patterns of the specific field.

Now, the transformers are acting as the backbone for many NLP tasks. On the MedNLI dataset, the Clinical-T5– Large model outperformed other models that doubled their parameters and achieved state-of-the art results. BERT-based models like MedBERT, DeBERTa, and other models that are pre-trained on domain-specific (biomedical) data are comparatively performing better than other models that are pre-trained on general domain data. The T5 (Text-To-Text Transfer Transformer) model will treat all tasks as text-to-text problems, and it performs well in tasks like text classification, question answering, and summarization. The adaptability of this model to various NLP tasks will be beneficial for analyzing medical texts.

3 Methodology

3.1 System Overview

For the NLI task, we will use the Flan-T5 model (which is an instruction-tuned T5 model). Based on its zero-shot performance on Template-1 in this task, the Flan-t5 model is chosen. In Flan-t5, there are 5 models from small (60 million) to xxl (11 billion) parameters. Among them, for this experiment, we have considered the base model and the large model. In this project, these models were subjected to zero-shot setting and fine-tuning using different instruction templates.

3.2 Dataset Pre-Processing

The NLI task has single and comparison types, where the 'Section_id' consists of eligibility criteria, intervention, results, and adverse effects. From these entries, statements are made; they make claims about a single CTR or compare two CTRs. In the case of the "single" type, all evidence will be contained in the primary CTR, while in the case of the "comparison" type, evidence will have to be retrieved from both CTRs in the same section. The evidence from the primary and secondary CTRs is combined to form the premise. The evidence for single type is passed as: *"Primary trial evidence are primary_evidence."* and for comparison type as: *"Primary trial evidence are primary_evidence and secondary trial evidence are secondary_evidence."*.

3.3 Instruction Template

We have taken a basic instruction template suitable for this task from the Flan-T5 template collection. Figure 1 shows the instruction template used for the model. The premise is replaced by the evidence mentioned above, and the hypothesis is replaced by the statement annotated by the domain expert. The options are replaced by *OPTIONS: Entailment* or *Contradiction*. This instruction is passed as input for both zero-shot and fine-tuned models.

{*Premise*}

Question: Does that imply that {*Hypothesis*}

{*Options*}

Figure 1: Instruction Template

3.4 Model Architectures

3.4.1 Zero-Shot Architecture

The zero-shot architecture shown in the Figure 2 illustrates a model for processing clinical trial reports (CTRs) on breast cancer using the Flan-T5 model. Initially, a dataset of structured CTRs is first divided into JSON-formatted training, development, and testing subsets. After preprocessing, that includes cleaning, normalization, and feature extraction, which yields a modified dataset optimized for analysis. An instruction template is guided into inputs of the modified data that are appropriate

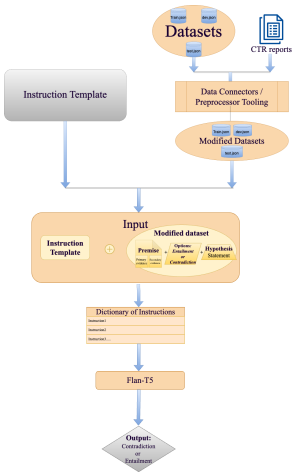

Figure 2: Zero-Shot Architecture

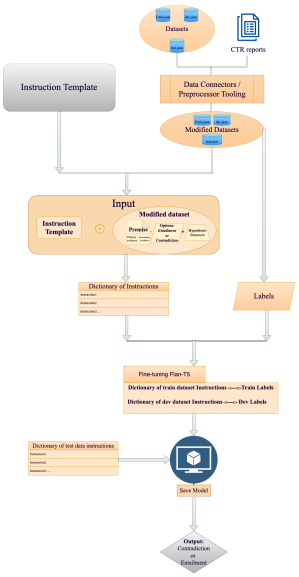

Figure 3: Fine-Tuning Architecture

for the model. Each input forms a premise and a hypothesis statement by combining the evidence with the dataset. The model evaluates two possible outcomes, i.e., entailment, where the hypothesis is supported by the premise, and contradiction, in which they are not. Inputs are now converted into a dictionary of instructions, where each is specifically designed to direct the Flan-T5 during the analysis. After receiving the instructions, the skilled natural language processor Flan-T5 evaluates the premise-hypothesis relationship. The outcome of each input is either an entailment or a contradiction as a result of this examination. The evaluation of CTRs is simplified by this model, which provides quick, automated insights into the vast data of breast cancer trials that might be crucial for clinical decision-making and research advancements.

3.4.2 Fine-Tuning Architecture

The fine-tuning architecture shown in the Figure 3 describes a machine learning model's workflow for analyzing clinical trial reports (CTRs) related to breast cancer. The process starts with raw datasets, which include CTR reports that provide the required context for data on breast cancer, as well as training, development, and testing JSON files. These datasets are transformed through data connectors and preprocessing techniques, resulting in modified datasets that are ready for analysis. The main step in the process is to combine the updated datasets with an instruction template to form structured inputs. Every input consists of a hypothesis statement and a premise that are sourced from either primary or secondary evidence. These are further divided into "Entilement" and "Contradiction" based on the relationship between the premise and the hypothesis. These inputs are structured according to a dictionary of instructions, which are then used to fine-tune the Flan-T5 model, which is a variation that is enhanced for understanding and producing text that is human-made text. Fine-tuning involves minimizing the loss function that quantifies the difference between "Entailment" or "Contradiction", a common loss function is the cross-entropy loss for classification.

Loss Function: It is defined for this classification problem as follows:

$$L(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \quad (1)$$

Here,

y is the actual label vector in one-hot encoded form.

$\hat{y}$ is the predicted probability distribution over the classes.

Now to minimize the loss function, the model parameters are updated using the gradient descent algorithm i.e

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta L(\theta) \quad (2)$$

Here,

η is the learning rate.

$\nabla_\theta L(\theta)$ is the gradient of loss function with respect to the parameters.

Backpropagation is used to calculaate the gradient of the loss function efficiently. This is important for training neural network architectures like Flan-T5. To prevent overfitting, L2 regularization

is added to the loss function. The modified loss function with regularization is:

$$L_{\text{reg}}(y, \hat{y}, \theta) = L(y, \hat{y}) + \lambda \|\theta\|^2 \quad (3)$$

Here,

λ is the regularization strength.

$\|\theta\|^2$ denotes the L2 norm of the model parameters.

Strategies for learning rate adjustment improves the convergence of the training i.e

$$\eta_t = \eta_0 \cdot e^{-\delta t} \quad (4)$$

Here,

η_0 is the initial learning rate.

δ is the decay rate.

t represents the training epoch.

Using the dictionary of labeled train and development dataset instructions, the model's parameters are iteratively adjusted as part of the fine-tuning process. The model is ready for deployment when it has been adjusted, and the results determine whether new theories are either contradicted by or entailed by the existing breast cancer CTR evidence. This AI-driven strategy aims to improve the decision-making process in breast cancer research and treatment by supporting the interpretation and validation of clinical hypotheses.

3.5 Evaluation Metrics

3.5.1 Generic Metrics

F1-score: F1 score is a measure of the harmonic mean of precision and recall. Commonly used as an evaluation metric in binary and multi-class classification and LLM evaluation, the F1 score integrates precision and recall into a single metric to gain a better understanding of model performance.

Mathematically,

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

3.5.2 Dataset Specific Metrics

Faithfulness: It is a measure of the extent to which a given system arrives at the correct prediction for the correct reason.

Mathematically,

$$\text{Faithfulness} = \frac{1}{N} \sum_{i=1}^{N} |f(y_i) - f(x_i)|$$

$$x_i \in C : \text{Label}(x_i) \neq \text{Label}(y_i), \text{ and } f(y_i) = \text{Label}(y_i)$$

Consistency: It is a measure of the extent to which a given system produces the same outputs for semantically equivalent problems.

Mathematically,

$$\text{Consistency} = \frac{1}{N} \sum_{i=1}^{N} |f(y_i) - f(x_i)|$$

where $x_i \in C : \text{Label}(x_i) = \text{Label}(y_i)$.

3.6 Software and Hardware

The total experiment is carried out using the T5 implementation of hugging face transformers. Flan-t5 base and large models are used for this experiment. The resources used for this experiment are colab pro with A100 GPU and personal computer.

4 Results

4.1 Results for flan-t5 (large) model

The training includes only train.json

Hyperparameters: Learning Rate(1e-4), Batch size =3 (fine-tuning), 6(zero-shot) Max-sequence length: fine-tuning-512, zero-shot - 5000.

Table 1: Performance Metrics for Zero-shot and Fine-tuning Models

Model (large)	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Zero-shot (f1-score)	0.67	0.66	0.62	0.67	0.66	0.67	0.66	0.67	0.68	0.65
Zero-shot (Faithfulness)	0.50	0.33	0.48	0.32	0.37	0.30	0.18	0.48	0.33	0.45
Zero-shot (Consistency)	0.56	0.47	0.53	0.48	0.50	0.49	0.43	0.54	0.49	0.51
Fine-tuning (f1-score)	0.61	0.62	0.62	0.59	0.51	0.51	0.66	0.66	0.52	0.32
Fine-tuning (Faithfulness)	0.25	0.28	0.08	0.49	0.45	0.40	0.05	0.11	0.52	0.72
Fine-tuning (Consistency)	0.46	0.49	0.40	0.45	0.49	0.49	0.40	0.43	0.50	0.56

From the obtained results, we can observe that the zero-shot model is performing better than the fine-tuning model. The reason is due to the batch size and maximum sequence length, because the sequence length used for zero-shot is 5000, whereas 512 for the fine-tuning model. Due to this, the fine-tuned model is only capable of reading instructions of less than 512. Increasing sequence length for the fine-tuning model results in memory issues. So, I have repeated the same process for the flan-t5 base model. As the size of the base model is smaller, we can increase the maximum sequence length.

From the above table, we can say that the models with templates T1, T2, T4, T5, T6, T9, and T10 are performing better than other templates.

4.2 Results for flan-t5 (base) model

The training includes only train.json

Hyperparameters: Learning Rate(1e-4), Batch size =3 (fine-tuning), 6(zero-shot) Max-sequence length: fine-tuning-2500, zero-shot - 5000.

Table 2: Performance Metrics for Zero-shot and Fine-tuning Models (Base)

Model (base)	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
Zero-shot (f1-score)	0.59	0.50	0.32	0.35	0.53	0.50	0.55	0.57	0.34	0.59	0.49
Zero-shot (Faithfulness)	0.42	0.66	0.84	0.84	0.43	0.56	0.43	0.55	0.73	0.44	0.41
Zero-shot (Consistency)	0.50	0.53	0.58	0.58	0.48	0.54	0.50	0.54	0.54	0.57	0.48
Fine-tuning (f1-score)	0.58	0.53	0.52	0.16	0.36	0.32	0.66	0.59	0.01	0.63	0.61
Fine-tuning (Faithfulness)	0.35	0.53	0.36	0.90	0.73	0.74	0.20	0.34	0.99	0.35	0.39
Fine-tuning (Consistency)	0.47	0.51	0.46	0.60	0.57	0.56	0.58	0.49	0.61	0.40	0.42

From the obtained results, we can observe that in templates T2, T3, T7, T8, T10, and T11, the f1 scores are high compared to the scores of zero-shot models. In templates T5, T6, and T4, the faithfulness and consistency of the fine-tuned model are better than zero-shot. This is because we have increased the sequence length in fine-tuning to 2500.

Upon comparing the models here, templates T2, T5, T6, T10, and T11 are performing better than other templates.

4.3 Results for flan-t5 base model (with data augmentation)

The training includes train.json + gold_practice_test.json

Hyperparameters: Learning Rate(1e-4), Batch size =3 (fine-tuning) Max-sequence length: fine-tuning-2500.

Table 3: Performance Metrics for Fine-tuning Model (Base)

Model (base)	T1	T2	T3	T4	T5
Fine-tuning (f1-score)	0.39	0.66	0.66	0.63	0.60
Fine-tuning (Faithfulness)	0.59	0.03	0.01	0.14	0.29
Fine-tuning (Consistency)	0.53	0.39	0.38	0.42	0.45

For data augmentation, we have joined the gold partice test.json file to the train.json file and used that as training data. Based on experimenting, the results are shown in the table above. This technique is performed on five templates, and from the results, we can observe that the model T5 is producing the best results among all. Even for the base model, we can observe that the f1 scores have now crossed 0.6. The reason is that the size of the training data has improved.

5 Discussion and Error Analysis

5.1 Results Analysis

- For large models, zero-shot models gave slightly better results than fine-tuned models. This might be because, for zero shots, we have provided a sequence length of 2500. But for fine-tuning, due to memory constraints, 512 is chosen as the maximum sequence length.
- When we look at the results of the base model, we can see that the scores of the fine-tuning models are higher in many cases than those of the zero-shot models. Because the sequence length used for the fine-tuning is 2500. Due to this, the results were improved.
- The large models are performing better than the base models. The parameters of the models have a huge difference. Which shows an effect on performance.
- Based on comparing all the results in all three methods, we conclude that Template 'T1' is performing better.

5.2 Comparison between milestone-2 and milestone-3

In milestone 2, we have used only one template, which is 'T1'. There is a change in the fine-tuning performance from milestone 2 to milestone 3.

Milestone	F1 Score	Faithfulness	Consistency
Milestone 2	0.549	0.38	0.47
Milestone 3	0.58	0.35	0.47

Table 4: Comparison of fine-tuning results of base model with template 'T1'

Upon experimenting with different hyperparameters, the performance improved. After an increase in performance, we considered 10 more templates and performed the operations. These results were included for milestone 3. We have also tried data augmentation and evaluated the performance of the fine-tuning models for different instruction templates.

5.3 Comparison of results with External existing Flan-T5 model

The above bar graph is from the flan t5 (base) model. We can infer from the above bar graph that our zero-shot model is performing better than the pre-existing model. In the case of fine-tuning,

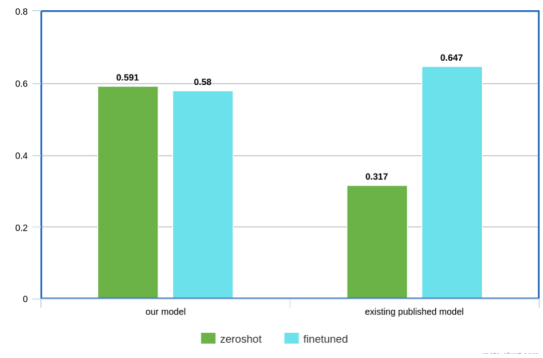

Figure 4: Comparison our (best) model with Existing Published model (both are flan-t5 base).

our model is producing an f1 score of 0.58, whereas the existing model produces 0.647.So, in the case of fine-tuning, the existing model is performing better than our model.

5.4 Comparison with few existing models other than Flan-T5

Comparing our proposed models i.e Flan-T5(large) and Flan-T5(base) models against the existing Tk-Instruct-11B and OPT-IML-30B with context of classifying statements in clinical trial reports (CTRs) as either entailment or contradiction the data presented.

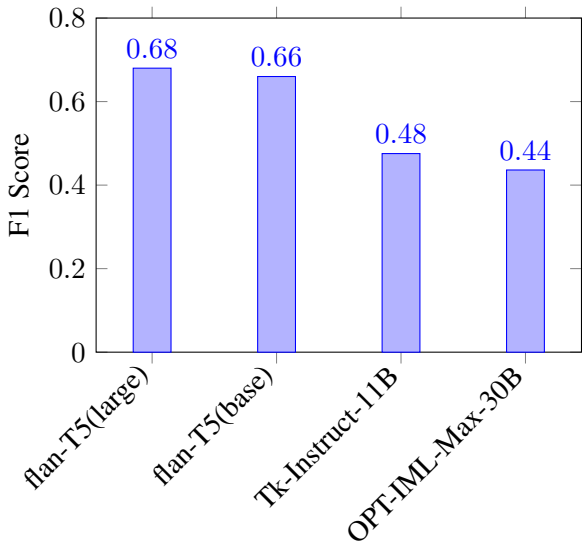

Figure 5: Comparison of Model Performances

Clearly for instruction template T7, the Flan-T5 (base) achieved an F1 score of 0.66,and for instruction template T9, the Flan-T5 (large) achieved an F1 score of 0.68 both of which are significantly higher compared to the scores of 0.48 and 0.44 for Tk-Instruct-11B and OPT-IML-Max-30B, respectively. These results shows the effectiveness of our models in more accurately and consistently classifying complex entailment and contradiction scenarios in clinical trial reports and validating our approach in improving the reliability and efficiency of automated report analysis in medical research.

5.5 Areas of improvement

Our fine-tuned model can be improved. Even though we applied many techniques, like using different templates, data augmentation, and hyperparameter tuning, to improve the performance of the fine-tuning models, We got better results compared to milestone 2. But the fine-tuned results can be improved, as we can see from the results for the existing model. Tried doing text summarization to reduce the size of premise, but the summarized texts obtained are not good, so haven't used text-summarization. The possible improvements are training Flan t5 with medical-related texts and vocabulary, trying different data augmentation techniques, and increasing batch size and sequence length if provided with suitable computational resources. Exploring and finding better hyperparameters.

6 Conclusion

In this project, we have used various instruction-tuned models using zero-shot settings and fine-tuning. The performance of the model depends on its size, i.e., as the size increases, the performance increases. We can observe that the performance of the model varies based on the instruction templates used. The best instruction template among the templates used is 'T1' in the case of zero-shot settings and 'T5' for fine-tuning models. In terms of models, the flan-t5 large with the 'T1' template under zero-shot is performing better among other zero-shot settings. In the case of fine-tuning the flan_t5 base model with the 'T5' template, it provides better consistency and faithfulness with a moderate f1 score. But after comparing all the results, we can say that the 'T1' template has a slight advantage over other templates considering all the results.

7 Limitations

In this project, It is constrained in using only the base and large variants of the Flan-T5 models due to resource limitations. This restriction has prevented from achieving optimal results, as larger models like Flan-T5 XL and XXL, which offer su-

perior learning capabilities, were not utilized. It is limited to smaller models due to computational resources, potentially impacting the optimal performance achievable with larger models. Used a max sequence length of 512 for fine-tuning due to resource limits, compared to 5000 for zero-shot experiments. This likely bettered the zero-shot approach, as longer sequences can hold more contextual data critical for complex tasks. Since the medical data is sensitive in nature our ability to train the models against different scenarios, potentially affecting their real-world relevance. Employed a random selection for hyperparameters, which may not be efficient. A more systematic approach could improve model performance and efficiency. Models overfitted when trained for higher epochs, particularly with data-augmented models, indicating a requirement for advanced regularization techniques and better training duration tuning. Models were tested only on specific instructional templates, potentially not showing their full capabilities across a broader range of tasks. Limited computational resources affected the number of experiments that could be conducted, impacting the depth of model exploration. By Improving resource allocation and using systematic model training approaches, exploring alternative data augmentation and hyperparameter tuning methods could help overcome these limitations in future research.

8 Instruction Templates

1. T1 - f"premise Question: Does this imply that hypothesis? options"
2. T2 - f"premise Question: Is it likely that hypothesis given the context?
3. T3 - f"premise Question: Based on the information, would hypothesis be supported? options"
4. T4 - f"premise Question: Does hypothesis align with the provided details? options”
5. T5 - f"premise Question: Would hypothesis logically follow from the given information? options”
6. T6 - f"premise Question: Can hypothesis be inferred from the paragraph? options”
7. T7 - f"premise Question: Is there evidence to suggest hypothesis is true/false? options"
8. T8 - f"premise Question: Does the paragraph suggest hypothesis is plausible? options”
9. T9 - f"premise Question: Would hypothesis be consistent with the context provided? options"
10. T10 - f"premise Question: Is there enough support in the text to conclude hypothesis? options"
11. T11 - f" Question: Can hypothesis be supported by premise? options"

9 Hyper Parameters

The hyper parameters used are as follows i.e

- 1e-4 3e-4 3e-5 3e-6 are the different learning rates
- 2,3,4,5,6 are the different batch sizes.

10 Contribution

Name	Contribution
Venkata Sai Vikas Katuru	Literature Review, Baseline architecure, zero-shot for 5 instruction templates, fine-tuning for 5 instruction templates, zero-shot architecture, Hyper-parameter tuning setup, code-refactoring, Completing few sections of report such as , Introduction, Related Work, few sections of methodology, Results, Bibliography,
Sai Satwik Dikkala	Literature Review, fine-tuning for 5 instruction templates, zero-shot for 5 instruction templates, fine-tuning architecture, README.md file creation, Completing few sections of report such as Abstract, Model architecures, Evaluation Metrics, Discussion and Error Analysis, Limitations, Conclusion

11 Bibliography

1.https://huggingface.co/docs/transformers/en/model_doc/flan-t5

2.https://huggingface.co/docs/transformers/en/model_doc/t5

3.https://huggingface.co/learn/nlp-course/en/chapter1/4

4.https://huggingface.co/docs/transformers/en/training

5.https://huggingface.co/tasks/zero-shot-classification

6.https://blog.futuresmart.ai/exploring-zero-shot-learning-and-huggingface-transformers-a-comprehensive-guide

7.https://aclanthology.org/2023.semeval-1.137/

8.https://sites.google.com/view/nli4ct/semeval-2024/dataset-description?authuser=0