Evaluation of General and Domain-Specifc STS Data by LLMs and beyond

Ameya Bhave, Rachit Shaha
AMEYAB1@UMBC.EDU, RACHITS1@UMBC.EDU

Abstract

The use of Large Language Models (LLMs) have made huge advancements in Natural Language processing (NLP) applications, from embedding techniques to generative text creation, it marks an exciting era in the NLP community. Opportunities have increased in exploring the ability and impact of different LLMs on different NLP tasks. Semantic text Similarity (STS) is one the areas which can be explored and can gain new benchmarks from the progress made in LLMs. These LLM models contain a wide variety and a broad overview of many topics and domains because they are pre-trained on large datasets with several parameters. This can become a drawback, as LLMs may lack the domain specific knowledge necessary for certain applications. Through this project we aim to study how LLMs perform on domain specific STS. Additionally we also want to find ways to improve LLMs' and understand its performance on domain centered data.

1 Introduction

Large language Models (LLMs) have demonstrated their potential in a wide variety of NLP tasks. They help in providing intial insights and understandings for users to assess, and comprehend various topics. The need for LLMs in bridging the gap for machines in handling complex language tasks such as translation, summarization, and information retrieval has been increasingly recognized (Naveed et al., 2023). Semantic Text Similarity (STS) has emerged as a significant area of interest on which LLMs can be used.

Early work by (Zhong et al., 2023), highlighted potential limitations in the suitability of LLMs, specifically ChatGPT, for the STS-Benchmark (STS-B) task. Their findings which were based on a small sample set (n=50) indicated that ChatGPT's performance was inferior compared to pre-trained RoBERTa models in STS applications. However, this perspective was challenged by subsequent studies, like (Gatto et al., 2023), who demonstrated that LLMs, including ChatGPT, could indeed be effective for STS tasks. Their comprehensive exploration involved evaluating ChatGPT and Llama2 across existing STS benchmarks and newly developed domain-specific STS challenge sets.

Such previous works form the backdrop for our current project, which is motivated by the need to further explore the capabilities of openly available LLMs for STS. Our approach is broadly divided into 2 tracks; firstly, we aim to assess the performance of these models like BERT, GPT2, XLNET, RoBERTa on general STS datasets, such as STS-12 to STS-16. This initial task is necessary to understand LLM capabilities in handling generic STS tasks. Secondly, we plan to use the best model on domain-specific STS datasets such as News, Sports, and Biomedical to understand the performance of LLMs on domain specific STS dataset. In addition to this objective we also wish to extend and explore potential customizations that can be made to LLMs to enhance their performance in these specific domains.

2 Proposed Solution

Our proposed solution addresses the challenges that may come with the use of Large Language Models (LLMs) for domain-specific Semantic Text Similarity (STS) datasets. The fact that LLMs are generally trained on broad, generalized datasets, may not be effective on the nuances of specific domains. To add on the capability of LLMs for domain-specific STS performance, we propose implementing customizable feature layers. These layers would be unique for each domain, with their weights being learned during the training process. Our methodology involves the following steps.

2.1 Benchmarking STS Datasets on Current LLMs:

We will begin by benchmarking pre-trained LLMs gainst general STS datasets. This step will help us understand the baseline performance of these models on STS tasks without customizations. We will assess which model is best suited for STS tasks based on the Spearmann correlation on these datasets.

2.2 Testing the Best LLM Model on Domain-Specific Data:

The LLM model that gives the highest score in the benchmarking step will be tested on domain-specific STS datasets. The aim is to evaluate whether this model maintains the scores in a domain-specific context as it did in the general STS benchmarking.

2.3 Customizing the Best Model for Domain-Specific STS Data:

To provide an enhanced model we propose adding one or more customizable layers to the best-performing LLM. These layers will perform linear transformations using the dense embeddings from the input sentences of the LLM models, along with the cosine similarity scores. These new layers will focus on learning weights for domain-specific STS instances, aiming to provide better score for the specific tasks.

3 Detailed Proposal

In our approach, we use well-known models like BERT, GPT2, RoBERTa, and XLNet. These models have been remarkable in recent NLP advancements and have been extensively tested in various contexts. Our choice to use these models aligns with its commonality within NLP research, ensuring that our work aligns with the NLP community practices. Hence the use of these models provides a direct link to the existing body of work, allowing us to compare and contrast our results with those from previous studies. (Devlin et al., 2019), (Liu et al., 2019), (Yang et al., 2020).

Our methodology incorporates Spearman correlation for testing, a statistical method widely used in NLP research for assessing the performance of models in STS task. This will ensure that our results are comparable with previous studies, many of which also rely on Spearman correlation for evaluating model performance. By adhering to this standard, our findings can be directly compared to existing research, giving a better understanding of the improvements or differences our solution offers. (Chen et al., 2016)

The key change in our approach is the implementation of customizable feature layers specific to each domain. This aspect of our solution is what sets it apart from previous work. While the foundational models we use (BERT, GPT2, RoBERTa, and XL-Net) have been explored in various capacities, the concept of adding domain-specific layers to these models for STS tasks is relatively unexplored. This customization allows for a more specific understanding and processing of domain-specific language which may lead to significant improvements in accuracy for STS tasks.

Previous research has often focused on the general applicability of models like BERT and RoBERTa across a broad range of tasks. Our work extends this by focusing specifically on the domain-specific challenges in STS. By adapting these models with customizable layers, we are exploring the possibility and methodologies of fine-tuning LLMs for specific domain needs. This not only has the potential to improve the performance of these models in specific areas but also contributes to the broader understanding of how flexible and adaptable these models can be.

4 Experimentation, Methodology

4.1 For Benchmarking STS Datasets on Current LLMs

Our experiment involves a systematic evaluation of several Large Language Models (LLMs) - specifically BERT, GPT2, RoBERTa, and XLNet - in the context of Semantic Text Similarity (STS) tasks. The core of our methodology is to assess these models using Spearman rank correlation coefficient. This approach is consistent with standard practices in NLP research and provides a reliable measure of model performance on STS tasks.

Experiment Setup

The experiment begins with the installation of essential Python libraries, including **transformers**, **evaluate**, and **datasets**. These libraries provide the necessary tools and pre-trained models for our analysis. We use the **datasets** library to load various STS datasets, specifically sts12-sts to sts16-sts and sickr-sts, for comprehensive testing across multiple years and contexts.

Model Evaluation

The function **getSpearmanCorr** is designed to calculate the Spearman correlation coefficient for a given dataset and model. This function takes the dataset and the name of the pre-trained model as inputs and follows these steps:

1. Tokenization and Model Loading: Each sentence pair from the dataset is tokenized using the tokenizer corresponding to the selected model. For models like GPT2, special consideration is given to tokenization details, such as setting the pad token.

2. Mean Pooling: After tokenization, we apply mean pooling to convert token embeddings into sentence embeddings. This step is crucial as it considers the attention mask for correct averaging of embeddings, ensuring that the model's understanding of each sentence is as accurate as possible.

3. Cosine Similarity Calculation: We then calculate the cosine similarity between the sentence embeddings of each pair. This similarity score gives us a measure for how similar the model sees the two sentences.

4. Spearman Correlation Computation: After processing the entire dataset, we calculate the Spearman correlation between the model's predicted similarity scores and the true scores provided in the dataset.

The function is then applied to each of the selected models (BERT, GPT2, RoBERTa, and XLNet) across all the chosen STS datasets. This approach allows us to compare the models' performance in a standardized manner.

4.2 Testing the Best LLM Model on Domain-Specific Data

From the results of the previous benchmarking experiment we saw that the BERT model gave a good score on the STS datasets and hence we use the BERT model for our next set of experiments.

In this phase of our research, we focus on testing the selected Large Language Model (LLM), specifically BERT, on domain specific STS datasets. This step will evaluate the model's performance in domain specific contexts and to assess its ability to handle the specificities of domain specific language.

Experiment Setup:

The experiment involves the following steps:

1. Importing and Preparing Datasets: We use three different domain-specific datasets - News, Sports, and Biomedical - to test the model's performance. The News and Sports datasets are read from CSV files, and the Biomedical dataset is imported from Huggingface Datasets. To ensure consistency, we format each dataset with uniform column names ('sentence1', 'sentence2', 'score'). The score is the Similarity score between the two sentences.

2. Model Evaluation: We use the **getSpearmanCorr** function to evaluate the model's performance on each dataset. This function follows a specific methodology:

2.1 Model Loading and Tokenization: We load the pre-trained BERT model and its tokenizer. Each sentence pair from the dataset is then tokenized.

2.2 Mean Pooling for Embedding Extraction: After tokenization, we apply mean pooling to the token embeddings, considering the attention mask for accurate averaging.

2.3 Cosine Similarity Calculation and Rescaling: For each sentence pair, we calculate the cosine similarity between their embeddings. The similarity scores are then rescaled from a range of (-1, 1) to (0, 1) to match the target scores in the dataset.

2.4 Spearman Correlation Computation: The final step involves calculating the Spearman rank correlation coefficient between the predicted similarity scores and the actual scores in the dataset.

4.3 Customizing the Best Model for Domain-Specific STS Data

In this phase, we focus on enhancing the Large Language Model (LLM), specifically BERT, with adding custom layers tailored for domain-specific Semantic Text Similarity (STS) tasks. This step is intended for improving model performance in specialized contexts.

Experiment Setup:

1. Data Preparation: We use the same three domain-specific datasets - News, Sports, and Biomedical. Each dataset is formatted and split into training and testing sets using train_test_split.

2. Model Definition: A custom model class, **BertForSTS**, is created. It utilizes a pre-trained BERT model and adds a fully connected linear layer for fine-tuning.

Model Training and Evaluation:

1. Custom Model Training

The **BertForSTS** model is instantiated and trained on each domain-specific dataset.

The model uses mean pooling to extract sentence embeddings, followed by a linear transformation

Model	STS12	STS13	STS14	STS15	STS16	SICKR	Average
BERT	0.3086968	0.5989485	0.4772789	0.6028566	0.6373272	0.5864510	0.5284
XLNET	0.3231934	0.246806	0.214138	0.371071	0.359946	0.381360	0.3116
GPT2	0.2584338	0.289084	0.262067	0.347398	0.356958	0.438282	0.3183
RoBERTa	0.3210866	0.563290	0.452196	0.613447	0.619760	0.629613	0.5283

Figure 1: Spearman Correlation results on implementing BERT, XLNET, GPT2 and ROBERTA on general STS datasets.

for fine-tuning. Training involves calculating a similarity score between sentence pairs, rescaling it, and using a loss function (MSE) to update the model's weights.

2. Spearman Correlation Calculation:
The trained model is then evaluated on the test set using a function **testgetSpearmanCorr**. This function calculates the Spearman rank correlation coefficient between the predicted similarity scores and the actual scores from the dataset.

5 Results

5.1 Results from the STS datasets for performance evaluation on different LLMs

From Figure 1 we understand the perfromance of LLMs on the genral STS Datasets and assess their performance by taking the Spearman correlation and then taking the average to know which model performed the best.
1. Analysis on the BERT Model:
Shows consistent performance across all STS benchmarks and SICK-R, with its lowest Spearman correlation on STS12 and the highest on STS16.
The average Spearman correlation across all the datasets is 0.5284, indicating a moderate to strong overall performance.
2. Analysis on the XLNET Model:
XLNETs' performance is more varied, with the highest Spearman correlation on STS12 and lowest in STS14.
The average performance of XLNET is 0.3116, which is significantly lower than BERTs'suggesting that XLNET might be less adept at capturing semantic similarities.
3. Analysis on the GPT2 Model: GPT2 also shows variability, with performance highest on the SICK-R dataset.
Its average performance is 0.3183, which is comparable to XLNET but still considerably lower than BERT.
4. Analysis on the RoBERTa Model:
RoBERTas' performance is quite consistent with BERTs', with its lowest performance on STS14 and the highest on STS16.
The average Spearman correaltion to RoBERTa is 0.5283, nearly identical to BERTs' average, suggesting similar overall capabilities in capturing semantic similarities across the datasets.

Model Name	SPEARMAN CORRELATION ON SPORTS DATASET	SPEARMAN CORRELATION ON NEWS DATASET	SPEARMAN CORRELATION ON BIOSSES DATASET
BERT	0.49926	0.25574	0.54698

Figure 2: Spearman Correlation results on implementing BERT on Domain Specific datasets.

5.2 Results from Testing the Best LLM Model on Domain-Specific Data

The Figure 2 gives the Spearman correlation coefficients resulting from experiments using the BERT model on three different domain-specific Semantic Text Similarity (STS) datasets.
1. Model Performance on Sports Dataset: The BERT model achieved a Spearman correlation coefficient of 0.49926. This suggests a moderate positive correlation between the model's output and the expected results, indicating that the BERT model had a fair ability to capture semantic similarity in the sports domain.
2. Model Performance on News Dataset: A Spearman correlation of 0.25574 indicates a weak positive correlation for the news dataset. This result implies that the BERT model struggled to understand and process semantic similarity in news-related texts compared to the other domains. The lower performance could be due to the complexity in news language.
3. Model Performance on Biomedical Dataset: With a Spearman correlation of 0.54698, the model shows a moderate correlation in the biomedical domain. This is the highest among the three, suggesting that BERT is relatively more effective at capturing the semantic nuances in biomedical texts.

5.3 Results from the Fine Tuning of Domain Specific STS

Figure 3 presents the loss calculated while training the model.

1. Analysis on Training Loss:
a. Sports Dataset: BERT exhibits a loss of 0.13146, which suggests that the model was relatively effective during training, fitting well to the sports domain data.
b. News Dataset: The model shows a loss of 0.183903, which is higher than the sports domain but still indicates a reasonable level of fit to the news domain data during training.
c. Biomedical Dataset: The loss spikes to 3.0323 for the biomedical dataset, which is substantially higher than the other domains. This indicates that the model had difficulty fitting to this specific domain during training, which could be due to the complexity and specificity of biomedical language.

Once the model is trained it is evaluated on the test dataset to get the Spearman Correlation result as shown in Figure 4.

2. Analysis of Spearman Correlation Coefficients
a. Sports Dataset: Post fine-tuning, the model's Spearman correlation drops to 0.25019, which indicates a weak positive correlation with the actual human annotations. This suggests that while the model had a low training loss, it did not generalize well to unseen data in the sports domain.
b. News Dataset: The model achieves a Spearman correlation of 0.37811, indicating a moderate correlation. Despite a higher training loss compared to the sports dataset, the model performed better on the unseen test data for the news domain.
c. Biomedical Dataset: After fine-tuning, the model shows an improved Spearman correlation of 0.58517. This is in contrast to the high training loss and suggests that despite the challenges during training, the model was able to capture semantic similarity in the biomedical domain effectively on the test set.
Overall Analysis :
The results indicate that the custom layers added to the BERT model had varying impacts on different domain-specific datasets. Notably, the biomedical dataset presented challenges during training but yielded the best performance during testing, suggesting that the fine-tuning allowed the model to capture domain-specific nuances that were not immediately apparent during training.
Conversely, the sports dataset, despite showing promise during training with the lowest loss, did not perform as well on the test data. This could imply overfitting to the training data or a mismatch between the training and testing data distributions. The news dataset showed a balanced outcome with moderate loss during training and a moderate Spearman correlation, indicating that the model's customizations were adequately generalized for this domain.

Model Name	LOSS ON SPORTS DATASET FOR TRAINING	LOSS ON NEWS DATASET FOR TRAINING	LOSS ON BIOSSES DATASET FOR TRAINING
BERT	0.13146	0.183903	3.0323

Figure 3: Initial Loss on training the new model on Domain Specific datasets.

Model Name	SPEARMAN CORRELATION ON SPORTS DATASET WITH FINETUNING	SPEARMAN CORRELATION ON NEWS DATASET WITH FINETUNING	SPEARMAN CORRELATION ON BIOSSES DATASET WITH FINETUNING
BERT	0.25019	0.37811	0.58517

Figure 4: Spearmann Corrleation on the new model for the Domain Speicfic STS

6 Limiatations

The limiatation of the work can be analysed on the following points.

6.1 Model Generalization and Overfitting

While the model trained well on the sports data, its generalization to the test set was less effective. This could point to limitations in the dataset diversity or size, and indicates a need for better regularization techniques or more robust cross-validation during training.

6.2 Dataset Complexity and Model Capability

The high training loss observed in the biomedical dataset suggests that the model struggled to fit the complexity of the data. This raises questions about the model's capacity to handle the specialized terminology and concepts intrinsic to biomedical texts. It may require deeper domain-specific layers or pre-training on biomedical literature to enhance performance.

6.3 Evaluation Metrics

While Spearman correlation is a robust metric for assessing monotonic relationships, it does not capture all aspects of model performance. Additional

metrics, such as precision, recall, and F1 score for different thresholds of similarity, could provide a more nuanced understanding of model efficacy.

7 Potential Future Works

7.1 Short Term

1. Model Regularization and Optimization: We may experiment with regularization techniques like dropout, weight decay, and early stopping to prevent overfitting, especially for datasets where the model performed well on training but poorly on testing.
2. Alternative Evaluation Metrics: We can introduce additional metrics like the Matthews correlation coefficient or F1 score to provide a more comprehensive evaluation of model performance.
3. Error Analysis: Perform detailed error analysis to understand the types of mistakes the model is making and identify specific areas for improvement

7.2 Long Term

1. Model Architecture Innovations: We need to research and develop novel neural network architectures that might be more suited to domain-specific STS tasks, potentially involving attention mechanisms.
2. Explainability and Interpretability: Focus on making the model's decisions more interpretable, which could involve techniques like Layer-wise Relevance Propagation (LRP) or SHAP values.
3. Ethical and Bias Analysis: Conduct thorough examinations of the models for biases, particularly given the domain-specific nature of the datasets, to ensure fair and ethical use.

References

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Joseph Gatto, Omar Sharif, Parker Seegmiller, Philip Bohlman, and Sarah Masud Preum. 2023. Text encoders lack knowledge: Leveraging generative llms for domain-specific semantic textual similarity.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert.