How similar are two LLMs: Comparative analysis based on embedding level.

Deepak Yadav
deepaky@iitg.ac.in
234156003

Mohd Adil Khan
adil.khan@iitg.ac.in
234156012

Reetu Raj Chauhan
c.reetu@iitg.ac.in
234156028

1 INTRODUCTION

The advent of Large Language Models (LLMs) has revolutionized the field of Natural Language Processing (NLP), enabling machines to perform complex language tasks with unprecedented accuracy. However, as these models become more advanced, the need for sophisticated evaluation metrics that can capture the depth and nuance of language understanding becomes increasingly critical. This project seeks to bridge this gap by introducing a nuanced approach to comparing semantic similarity between two LLMs, leveraging the tasks of fill-in-the-blanks and extractive question answering.

2 PROBLEM BEING SOLVED

At the heart of this project lies the challenge of measuring semantic similarity—a task that is inherently complex due to the multifaceted nature of language. Traditional metrics, such as the F1 score, are limited in their ability to capture semantic nuances, as they focus on exact lexical matches. This limitation becomes particularly evident when LLMs generate responses that are incorrect yet semantically similar to the expected answer. The project addresses this challenge by proposing a method that evaluates the semantic content of the models' responses, rather than just their lexical accuracy.

3 IMPORTANCE.

Semantic similarity is a fundamental aspect of NLP, impacting a wide range of applications from information retrieval and text summarization to machine translation and question answering. By focusing on semantic rather than lexical similarity, this project contributes to the creation of more intelligent, context-aware, and human-like NLP systems.

Benchmarking Language Models: When comparing and evaluating different LLMs, it becomes crucial for researchers, developers, and organizations to leverage these models effectively across various NLP tasks. Our approach offers a systematic method for benchmarking LLMs by assessing their performance on a standardized task.

Understanding Model Capabilities: By evaluating LLMs on the Fill in the Blanks task, we gain insights into their ability to understand context, infer missing information, and generate accurate and coherent responses. The application on Extractive question answering helps in evaluating similarity based on the capability of the model to learn and retrieve information from a given context in zero-shot scenario. This understanding aids in choosing the most suitable model for specific NLP applications and tasks.

Quality Assessment: Assessing the quality of the generated text is essential for ensuring the reliability and trustworthiness of LLMs, particularly in applications where the generated text is used for decision-making or communication with users. Our approach enables objective evaluation of the quality of text generated by different LLMs.

Improving Model Performance: Analyzing the similarities and differences between the answers generated by different LLMs provides valuable feedback for model improvement and fine-tuning. Understanding where and why models diverge or converge in their responses can guide future developments in LLM architecture, training, and fine-tuning strategies.

Application to Downstream Tasks: The Fill in the Blanks task serves as a proxy for various downstream NLP tasks, such as question answering, text completion, and language understanding. Evaluating LLMs on this task can provide insights into their performance on more complex tasks and their generalizability across different domains and contexts.

Identifying Illegal model reuse: Assistance in detecting illegal model reuse by measuring the similarity of representations, which can also shed light on the inherent similarities between different LLMs

4 PAST AND RELATED WORKS

Computing the similarity of two large language models (LLMs) is a complex task that involves understanding and comparing the representations and behaviors of the models. Some methodologies that researchers use to evaluate the similarity between LLMs are:

Representation Similarity Analysis (RSA): This method involves comparing the internal representations (activation patterns)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2024 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of neural networks when processing the same input. It's used to determine how similarly two models process information. [2]

Behavioral Testing: By comparing the outputs of LLMs on a standardized set of tasks or prompts, researchers can infer the similarity in their language understanding and generation capabilities.

Transfer Learning Performance: Evaluating how well a pretrained model adapts to new tasks can provide insights into the similarities in their learned representations.

Probing Tasks: These are diagnostic tasks designed to investigate what kind of linguistic information is encoded in the models' representations.

Similarity-Based Prompt Construction: This method involves designing prompts that leverage the memory capabilities of LLMs to format outputs based on knowledge from previous conversations or tasks.[3]

5 PROPOSED SOLUTION

In order to compare two Language Models (LLMs), we devised a methodology centered around the Fill in the Blanks task. Our approach involved generating answers using both models and subsequently calculating the cosine similarity between these answers.

5.1 Cloze Filling Task.

Data Collection and Preprocessing: We curated a dataset from the "SimpleBooks: Long-term dependency book dataset with simplified English vocabulary for word-level language modeling" to facilitate the Fill in the Blanks task. Leveraging this text corpus containing short stories, we tailored a dataset specifically for our task. Preprocessing steps involved removing extraneous expressions from the stories, as well as delimiters such as end-of-line and end-of-story markers. Additionally, we filtered out sentences of shorter length, ensuring each sentence contained only a single masked token.
Model Implementation: Utilizing RoBERTa[4] and T5[5], we generated answers for the fill-in-the-blank questions derived from our curated dataset.
SemScore Analysis: As the answers generated by the T5 model were multi-word, we conducted SemScore analysis to gain a better understanding of the two LLMs. This comprehensive approach allowed us to evaluate and compare the performance of RoBERTa and T5 in the context of the Fill in the Blanks task. It provided valuable insights into their respective capabilities and potential applications.

5.2 Extractive Question Answering.

The idea is to compare large language models (LLMs) on the responses they have generated for extractive question-answering tasks. In the Extractive question answering (EQA) task, we give the model context and question to be answered in context as a prompt, and the model gives us responses back from the context.

Example: prompt : [question = "Why is sky blue, precisely"context = "The sky appears blue due to the scattering of sunlight off the atmosphere. The Earth's atmosphere is composed of various gases and particles. When sunlight hits the atmosphere, shorter blue wavelengths are scattered in all directions by the gases and particles, making the sky appear blue to our eyes."]. Expected response :["due to scattering of the sunlight of the atmosphere"]

The responses from any two models for similar EQA prompts can be compared using some metric and reported as a similarity measure.

Here, we have used the F1 score, Exact Matches, and SEMSCORE[1] as metrics for comparisons.

6 EVALUATION

For the calculation of the metric F1 score, we employed two methods. In method 1, both the gold standard (actual) and predicted responses are first tokenized into words based on whitespace separation. Then, the function calculates the F1 score by considering the precision and recall of the predicted tokens against the gold tokens. Method 2 tokenizes the gold and predicted responses into individual characters, including alphabets and symbols, creating 1-gram. This method focuses on the similarity at a character level rather than a word level. The rationale behind this approach is to capture finer-grained similarities between the responses gathered through the respective tasks.

Exact Matches (EM) are instances that give exact responses as the gold standard; in our work, we have taken instances as True EM when the F1 score with both methods is equal and is 1. The Rationale is that sometimes, method 1 gives zero response. For instance, if the gold response is "Remarkable" and the model response is "Remarkably," method 1 gives a 0 F1 score, but method 2 gives a reasonable score. However, Method 1 is more logical as taking words as tokens is more reasonable than counting the number of similar characters in the response. So, it's a tradeoff, but looking at the minimum exact matches, it's a perfect method.

SEMSCORE is a cosine similarity metric based on the paragraph embeddings of the all-mpnet-base-v2 sentence transformers embeddings. We take the model response for the task and gold standard, get the sentence embeddings for both, take dot products of normalized embeddings, and get the cosine similarity. A score of 1 is a perfect match, and -1 is the worst match. The code and datasets can be viewed at this link of the repository on GitHub.

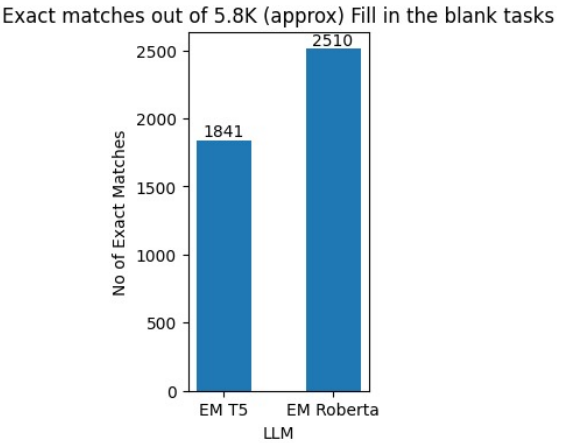

Figure 1: Number of Exact Matches with target fill word and model generated word.

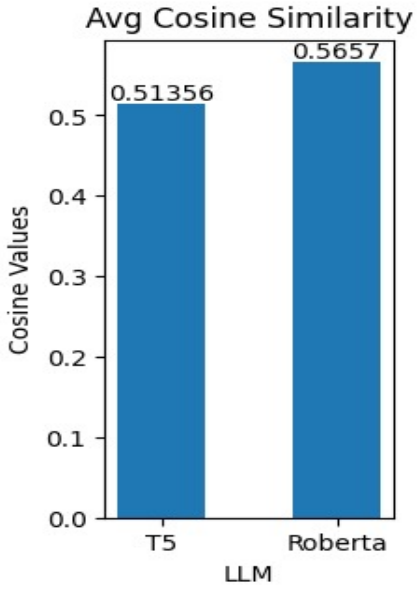

Figure 2: Average Cosine similarity between the target fill in word and model generated words for T5 and Roberta for cloze filling task.

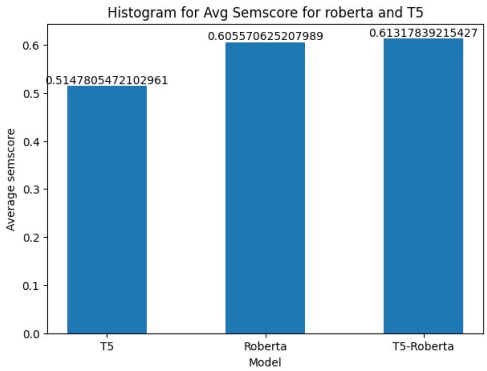

Figure 3: Average SEMSCORE similarity for target word and model response,and T5 reponse vs Roberta response for cloze filling task.

6.1 Cloze Filling tasks:

In our study, we conducted the Cloze filling task on a dataset consisting of 5.8k data points. We observed that T5 achieved an exact match on 1841 data points, while RoBERTa matched on 2510 data points. Subsequently, we calculated the average cosine similarity for both models. The T5 model exhibited an average cosine similarity of 0.51356, whereas RoBERTa demonstrated a slightly higher value of 0.5657. Despite T5 being considered a superior model, its performance was subpar. This discrepancy could potentially be attributed to T5 generating multi-word answers. To delve deeper into this issue, we performed SemScore analysis on the answers generated by both models. The SemScore between T5 and the exact answer was 0.5147, whereas the SemScore between RoBERTa and the exact match was 0.6055. Furthermore, the SemScore between T5 and RoBERTa was 0.6131, indicating a higher similarity between their answers. This suggests that although T5 and RoBERTa may provide incorrect answers, their responses are similar, possibly due to their underlying similarities in language understanding. These findings shed light on the performance and behaviour of T5 and RoBERTa in the Fill in the Blanks task, highlighting nuances that may affect their practical application in real-world scenarios.

6.2 Extractive Question Answering tasks:

For the Extractive question-answering task, we have used the SQuAD 2.0 dataset. SQuAD 2.0 is a widely used dataset for machine comprehension tasks in natural language processing. It consists of over 130K context-question-answer pairs, challenging models with unanswerable questions intentionally included. We have used models T5-base and Roberta-base to generate over 100K EQA prompts. After refining the responses, we have 50K data points with context-question-answer, responses by Roberta, and responses by T5. Treating answers as the gold standard, we evaluated the F1 score, Exact Match (boolean), and SEMSCORE for 50K responses from Roberta and T5. Figure 4 shows the number of exact matches with the target response by the models. Out of 50K instances, T5-base gave 26.562K responses, exactly the same as the target, whereas Roberta gave 25.442K exact matches. It seems the T5-base is better than Roberta for the EQA task. However, Figure 5 shows instances of zero F1 scores, representing no similarity in the responses with the target response. We observed that 1,468 responses by T5 and 1,258 by Roberta are vague/irrelevant. So, nothing concrete can be concluded from here. A high SEMSCORE between two texts denotes that the text is similar; responses with high similarity, close to 1 (say 0.9), may be considered good/acceptable responses. In Figure 6, we can see 28.991K responses by T5 have SEMSCORE similarity of 90% or greater, and for the Roberta model, we have 27.845K instances of the same. Interestingly, There is an almost similar score when the SEMSCORE is computed between the response of Roberta and T5; we see 28.662K instances where the responses have 90% or more similarity, indicating that both models have very similar responses for 57.324% (28.662K/50K) of total responses. Interestingly, in Figure 7 when we plot the histogram of all the data points with an SEMSCORE similarity of less than 90%, we see a stark similarity between distributions for both models. It should be noted that less similarity represents bad responses. So, both models

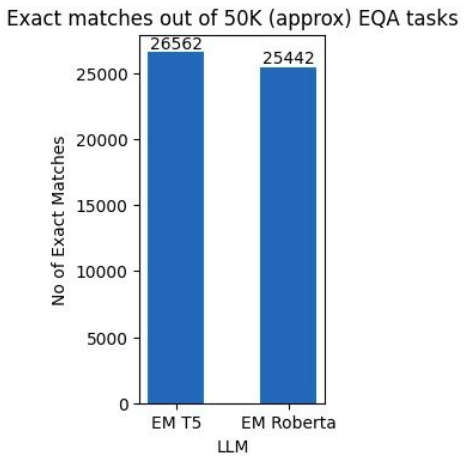

Figure 4: Number of Exact Matches with gold answer and model generated response.

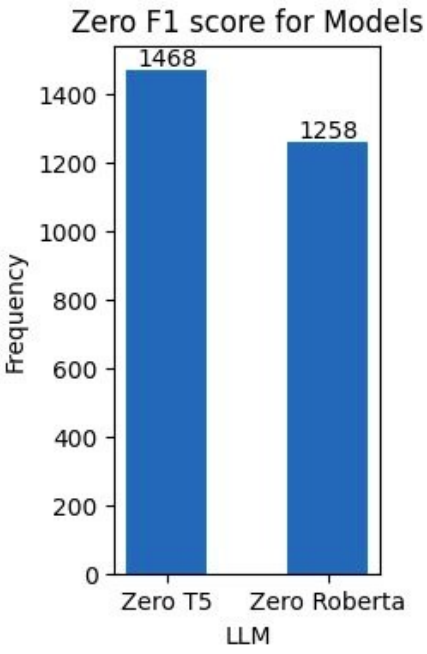

Figure 5: Number of instances with zero F1 score for model generated response and gold answers.

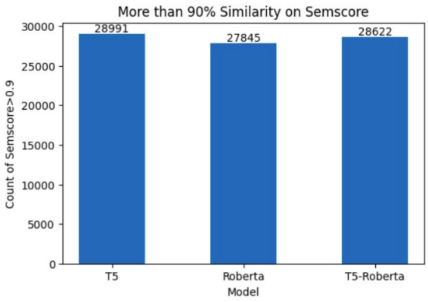

Figure 6: Number of instances with greater than 90% similarity on SEMSCORE for each model vs gold response, model vs model.

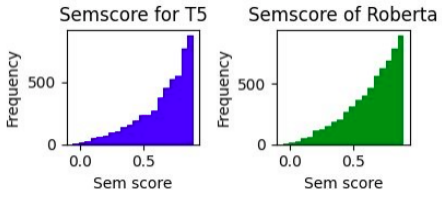

Figure 7: Histogram Plot for model instances with less than 90% similarity on SEMSCORE

have high similarities not only in good responses but also in bad responses.

7 LIMITATIONS

In our study, we encountered several limitations that are important to acknowledge:

Larger models like T5 can generate multi-word answers for the Fill in the Blanks task. However, this poses a challenge when implementing cosine similarity, as the comparison of multi-word answers becomes more complex.

Larger models tend to retain context across multiple Fill in the Blanks tasks, which can influence the generated answers. This context dependency makes the answers dependent on the sequence of Fill in the Blanks tasks, potentially affecting the accuracy of the evaluations.

While RoBERTa is specifically designed for tasks like Fill in the Blanks and performs well in such scenarios, T5 may not be optimized for this specific task. As a result, T5 may be restricted in its ability to accurately answer Fill in the Blanks questions, impacting the comparability between the two models. Models differ in architectures, training instances, task-specific design choices, and fine-tuning. Evaluating the similarity between models for a particular task may get biased results, and thus, it cannot be treated as some standard of similarity.

This work explored task-specific similarity between models for two NLP tasks. Other ways to quantify the similarity between the models can be based on architectures, training data, hyperparameters, and fine-tuning processes. Which can be taken as future work.

Humans have a very strong ability to make judgments about the writer based on just reading responses to very few specific questions; treating models as intelligent things means they should be open to critical judgment based on the corpus they generated. This unexplored domain needs to be picked up in future works.

Acknowledging these limitations is essential for interpreting the results of our study accurately and understanding the factors that may affect the performance of different language models in NLP tasks.

REFERENCES

[1] Ansar Aynetdinov and Alan Akbik. 2024. SemScore: Automated Evaluation of Instruction-Tuned LLMs based on Semantic Textual Similarity. *arXiv preprint arXiv:2401.17072* (2024).

[2] Max Klabunde, Mehdi Ben Amor, Michael Granitzer, and Florian Lemmerich. 2023. Towards Measuring Representational Similarity of Large Language Models. *arXiv preprint arXiv:2312.02730* (2023).

[3] Gaofei Liu, Meiqi Pan, Zhiyuan Ma, Miaomiao Gu, Ling Yang, and Jiwei Qin. 2023. Similarity-Based Prompt Construction for Large Language Model in Medical Tasks. In *China Health Information Processing Conference*. Springer, 73–83.

[4] 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. *arXiv preprint arXiv:1907.11692* (2019).

[5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.