Benchmarking ChatGPT

We tested OpenAI's ChatGPT for investing and compared it to our deep learning model ensemble, which features 27 intrinsic value metrics and a Mixture of Experts (MoE) approach. We'll dive into the benefits of generative AI, RLHF, and share our experimental results.

Generative AI is changing the expectation for financial analysis. With its ability to understand complex patterns and generate human-like text, it can quickly provide valuable insights and predictions. It brings model explainability to a whole new level. How did it stack up numerically?

We did a deep dive into the literature on Reinforcement Learning from Human Feedback, or RLHF for short. There's a detailed 2020 NeurIPS article Learning to Summarize from Human Feedback paper cited by the ChatGPT blog post that explains the core technology.[1] While there are great videos out there explaining RLHF, we noticed that they overlook a few essential points.

RLHF teaches a computer system, known as an agent, how to make better decisions in various situations. The main idea is to guide the agent by providing feedback on its actions, so it learns from experience and gradually improves its performance. To explain RLHF, we can break it down into two main components: the reward model loop and the PPO optimization.

Reward Model Loop:

The approach from Open AI uses a helper model, the reward model RM, to select between a generated output and human text. The loop can be broken into steps like so:

a. Collect data: The language model generates responses to a variety of input prompts or questions.

b. Rate data: Human experts review the generated responses and human responses and rate them based on their quality, relevance, and other factors.

c. Train reward model: Using the expert ratings, a reward model is trained to help the language model understand which responses are more desirable. The reward model calculates a numerical value which is the reward given to the reinforcement learning model.

PPO Optimization:

Proximal Policy Optimization (PPO) is used to fine-tune the model's response generation process. The goal is to help the model generate responses that result in higher rewards (meaning better ratings from human experts) while avoiding sudden changes in its behavior that could cause instability or other undesirable effects.

In PPO, we fine-tune the language model to generate better responses. The model takes a prompt and returns a sequence of text, and its main goal is to maximize the reward function, which measures how good the generated responses are.

Part 1 – Problem formulation: The language model generates text in response to input prompts. It has a vast action space, consisting of all possible words or tokens, and an observation space that includes various input sequences. The reward function combines the reward model and a constraint on policy shift to guide the model's improvement.

Part 2 – Implement reward: For a given input prompt, two texts are generated - one by the initial language model and the other by the current iteration of the fine-tuned model. The reward model evaluates the text from the current policy, and the difference between the two texts is calculated using a penalty, like the KL divergence. The final reward is the combination of the reward model evaluation and the penalty term.

Update the model: The PPO algorithm is used to optimize the language model's parameters, maximizing the reward metrics in the current batch of data. PPO ensures that the updates don't destabilize the learning process by using constraints on the gradient, the numerical model changes.

Iterate and improve: The RLHF process can be further enhanced by iteratively updating the reward model and policy together. As the policy updates, users can continue ranking the outputs, leading to complex dynamics between the policy and reward model evolution.

By iterating through the reward model loop and applying PPO optimization, the language model can gradually improve its performance, becoming better at understanding user inputs and providing more accurate, relevant, and useful responses. This process helps to create more advanced and user-friendly AI systems that can effectively assist users in various tasks and applications.

What makes RLHF so powerful? By combining reinforcement learning with generative GPT-based architectures, every token is subject to log probability estimation, meaning human feedback trains the neural network at every single token!

Imagine you're listening critically to speaker hanging on every single word uttered. One single wrong word can cause a total disconnect with the speaker and the message. This is an experience all humans have listening to a speech and now RLHF has recreated it.

Every word matters – that's the level of precision RLHF brings to the table. We believe it holds incredible potential with fine-grained token-by-token training.

We wanted to see how ChatGPT would fare against our model ensemble, so we put it to the test. We crafted a user prompt to get ChatGPT to generate financial analysis, and the results were quite impressive. Specifically we wanted the model to immediately give us a classification grade score and a confidence level from 0 to 1. After a little bit of prompt engineering, we were able to mostly get the output to conform. We extracted the grade and confidence score using a regex. We noticed sometimes the API would be overloaded and another technical issue was the total amount of tokens for an API call was limited to 4097 tokens for the GPT-3.5 turbo model we tested against. Let us know if you want more details on the technical points or prompt, and we'd be happy to share. Here are a few of the qualitative results, outputs from one of the better GPT models.

The first one is under the symbol MKTX.

```
Grade: B. Confidence: 0.8. Market Axess Holdings Inc. has a strong business model with a leading electronic
trading platform in fixed-income markets. The company has consistently paid dividends and has authorized
multiple share repurchase programs. However, the lack of information on intrinsic value metrics such as free
cash flow yield and profit margin makes it difficult to give a higher grade.

Grade: C. Confidence: 0.6. The company's financial filing shows mixed results across its segments. While the
Securities Services and Market and Wealth Services segments have strong revenue and pre-tax operating
margins, the Investment and Wealth Management segment has a low pre-tax operating margin. Additionally, the
company's pension and healthcare benefit obligations are significant and may impact future cash flows.
However, the company's overall financial position is stable, with consistent revenue and net interest
revenue.
```

Grade: B. Confidence: 0.8. Market Axess Holdings Inc. has a strong business model with a leading electronic trading platform in fixed-income markets. The company has consistently paid dividends and has authorized multiple share repurchase programs. However, the lack of information on intrinsic value metrics such as free cash flow yield and profit margin makes it difficult to give a higher grade.

Grade: C. Confidence: 0.6. The company's financial filing shows mixed results across its segments. While the Securities Services and Market and Wealth Services segments have strong revenue and pre-tax operating margins, the Investment and Wealth Management segment has a low pre-tax operating margin. Additionally, the company's pension and healthcare benefit obligations are significant and may impact future cash flows. However, the company's overall financial position is stable, with consistent revenue and net interest revenue.

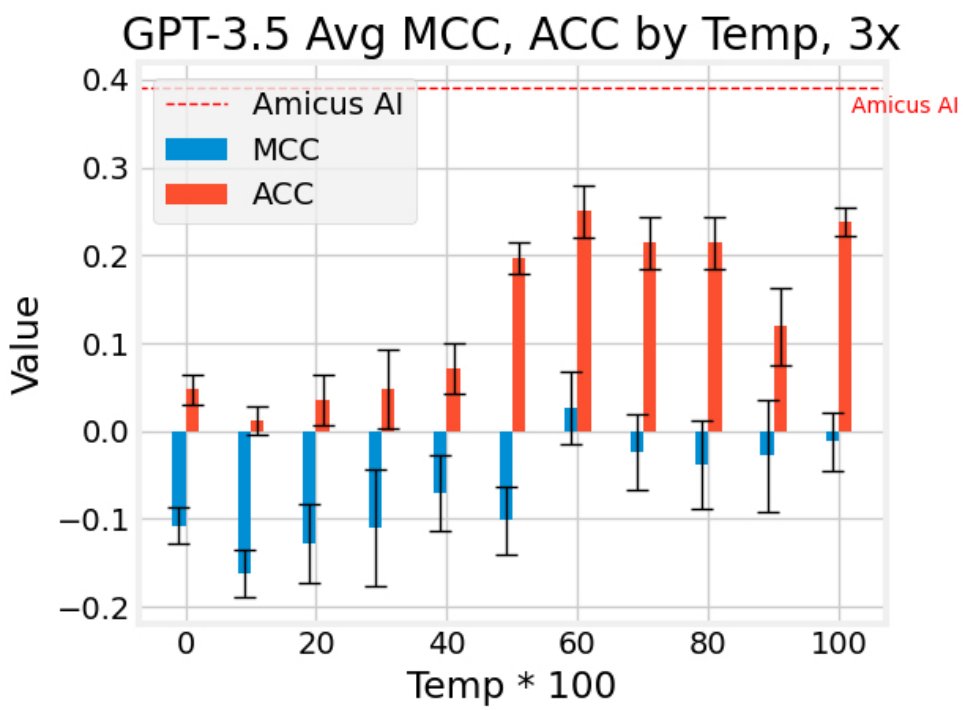
Those are impressive samples. To test ChatGPT, we plugged it into our evaluation framework, which uses a train/validation/test structure that's essential in machine learning. This ensures that our model performs well on unseen data and controls against cheating and overfitting. And we discovered that ChatGPT's potential varies wildly with one critical parameter – the temperature, which affects the output randomness.

Experts in machine learning can skip the next section but it's worth reiterating. The train, validation, and test structure is a crucial approach in developing effective machine learning models. It involves splitting the dataset into three parts: training which the bulk at 80% of the dataset, validation partition at 10%, and testing at 10%. This structure is effective for three main reasons:

- Preventing overfitting: Using separate datasets for training and evaluation minimizes the risk of overfitting. The model is trained on the training set and evaluated on the validation set, ensuring it performs well on unseen data.
- Model selection and hyperparameter tuning: The validation set allows for the comparison of different models and tuning of hyperparameters, helping to find the best model configuration without influencing the final evaluation on the test set.
- Unbiased performance estimation: The test set provides an independent assessment of the model's performance on unseen data, ensuring that the evaluation is not biased by decisions made during training and hyperparameter tuning.

Our case: We have around 500 companies, meaning 450 texts for training, 50 for validation, and 50 for testing. Normally, we train the model using the 450 samples, evaluate and tune the model with the validation set, and finally, assess the model's performance using the 50-sample test set. This approach ensures the model is less likely to overfit and provides a reliable estimation of its performance on new, unseen data. For our in-house model which is at the product level, we've optimized and frozen the model hyperparameters and use the validation set only for model selection. In our comparison, we're comparing the test set performance of our model vs GPT 3.5 turbo. And that's when we discovered that the key to unlocking ChatGPT's potential lies in the temperature.

The temperature appeared incredibly important for text generation. In OpenAI's words, "What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic." For instance, when we set the temperature to less than 0.2, the ratings experienced mode collapse, where they would only issue the same rating, letter grade C, to all companies.



We swept the temperature parameter from 0 to 1 and plotted the Matthews Correlation Coefficient (MCC) and accuracy of the test results. The MCC is a particularly important metric. MCC is a statistical metric used to measure the quality of classification models. It takes into account true positives, true negatives, false positives, and false negatives to provide a single score that indicates how well the model is performing. The MCC value ranges from -1 to +1, where a value of +1 represents a perfect prediction, 0 represents a completely random prediction, and -1 represents complete disagreement between the predicted and actual labels. In general, MCC values greater than 0.1 are considered to be good, while values less than 0.1 indicate poor performance.

The MCC can be a more informative measure of model performance than traditional metrics like accuracy, which can be misleading if the model is biased towards the majority class. Our model ensemble outperformed all experiments. We repeated each test for a given temperature 3 times and plot an error bar for the standard deviation to give an idea of the stability. During the tests, we used a whopping 6.35 million tokens across input and completion!

Here is the figure summarizing the results. On the horizontal we have increasing temperature from 0 to 1, which practically allowed different ratings at higher ends. On the vertical, we have the MCC and accuracy. We can see that they have a rough correlation- a higher MCC will naturally have a higher accuracy. We'd expect a MCC of 0 to be equivalent to random chance which would imply an accuracy of 20% for quintiles. On the chart we can find the best GPT temperature setting was 0.6 which gave 25% accuracy or 5% above random chance. The corresponding MCC value was

0.026. We can compare to an Amicus model which had 39.1% accuracy or 57% greater accuracy than the best GPT model.

What's not shown on the chart is our ensemble has a MCC value on the order of 0.23 which is a step function above the MCC values from GPT, at least 874% higher. Having a higher MCC value implies a much better grasp of value concepts across all types of businesses.

Despite not outperforming our ensemble, generative AI is exciting. ChatGPT showcased explainability and generated confidence levels for its predictions. While we didn't analyze those levels, we see huge potential for future applications. One possible downside is RLHF has been known to distort the calibration of confidence levels. OpenAI corroborates our experience that confidence levels prior to RLHF are relatively calibrated.

When, we required the model to give a short explanation of their rating, the better models had impressive thought-provoking analysis. At the very least, the AI explanations are good for brainstorming and out of the box thinking.

Limitations & Future Prospects

It's important to note that we were limited to 4097 tokens for the GPT 3.5 turbo model (a close cousin of ChatGPT), while our models read up to the required 200k tokens per company. We also didn't use the more advanced GPT-4, which supports longer context up to 32k tokens, but at a much higher inference cost and time. GPT has a natural user interaction, and RLHF has an even more enticing prospect.

We're crafting our own dataset. Collecting a dataset for expert work is a challenge that can't be crowdsourced like OpenAI's approach to GPT-4 data. The fact is even 90% of professionals underperform the market. Humans have a tough job with the difficulty of overcoming biased research and news that inundate investors. We aim to combine top level human feedback with RLHF techniques to elevate our models to unprecedented heights in the service of an unbiased, quantitative service for investors.

If you have any questions or want more information on our methods, don't hesitate to contact.

1. Learning to summarize from human feedback: https://arxiv.org/abs/2009.01325v2

2. Mixture of Experts: https://arxiv.org/abs/1701.06538