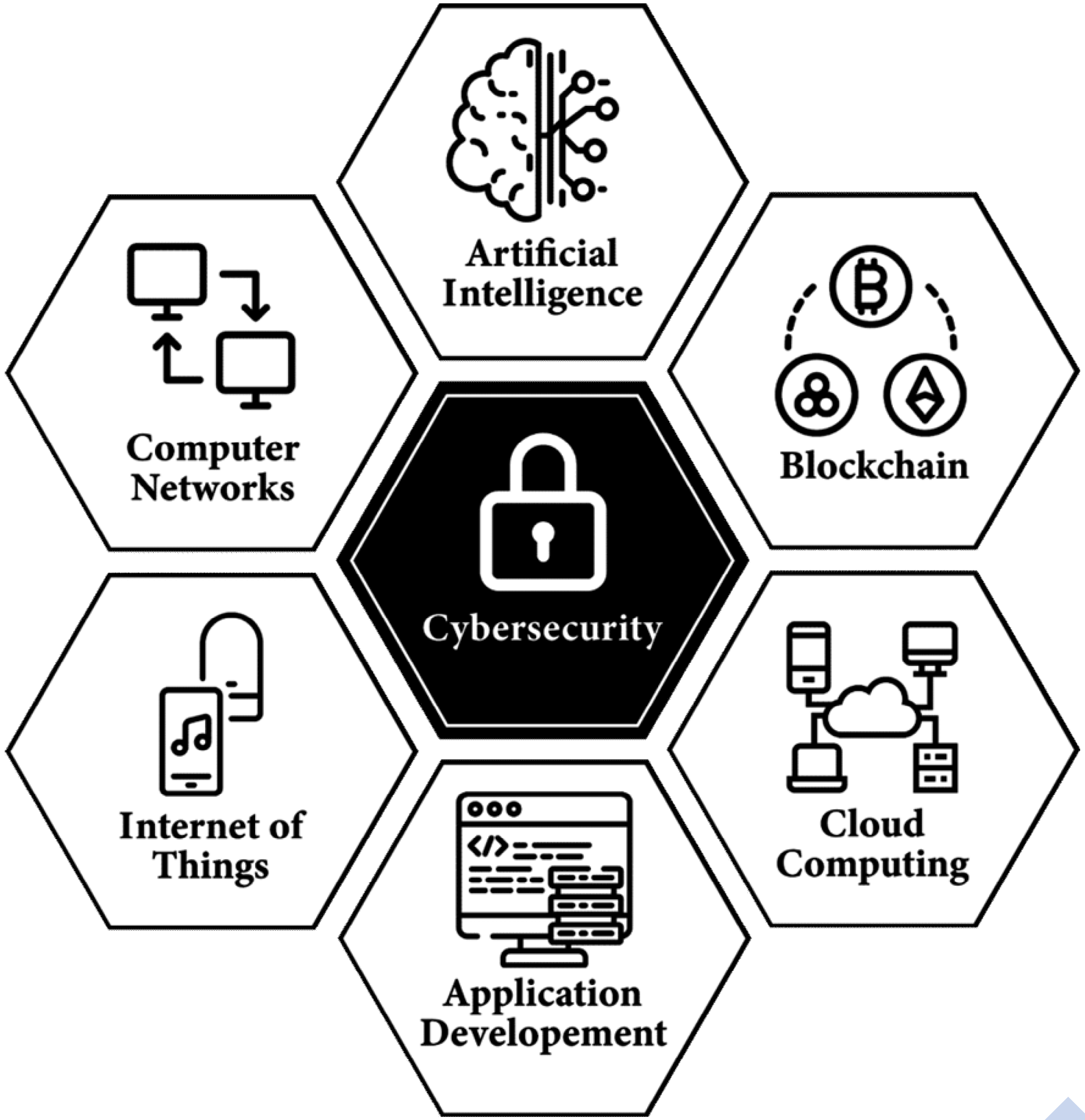
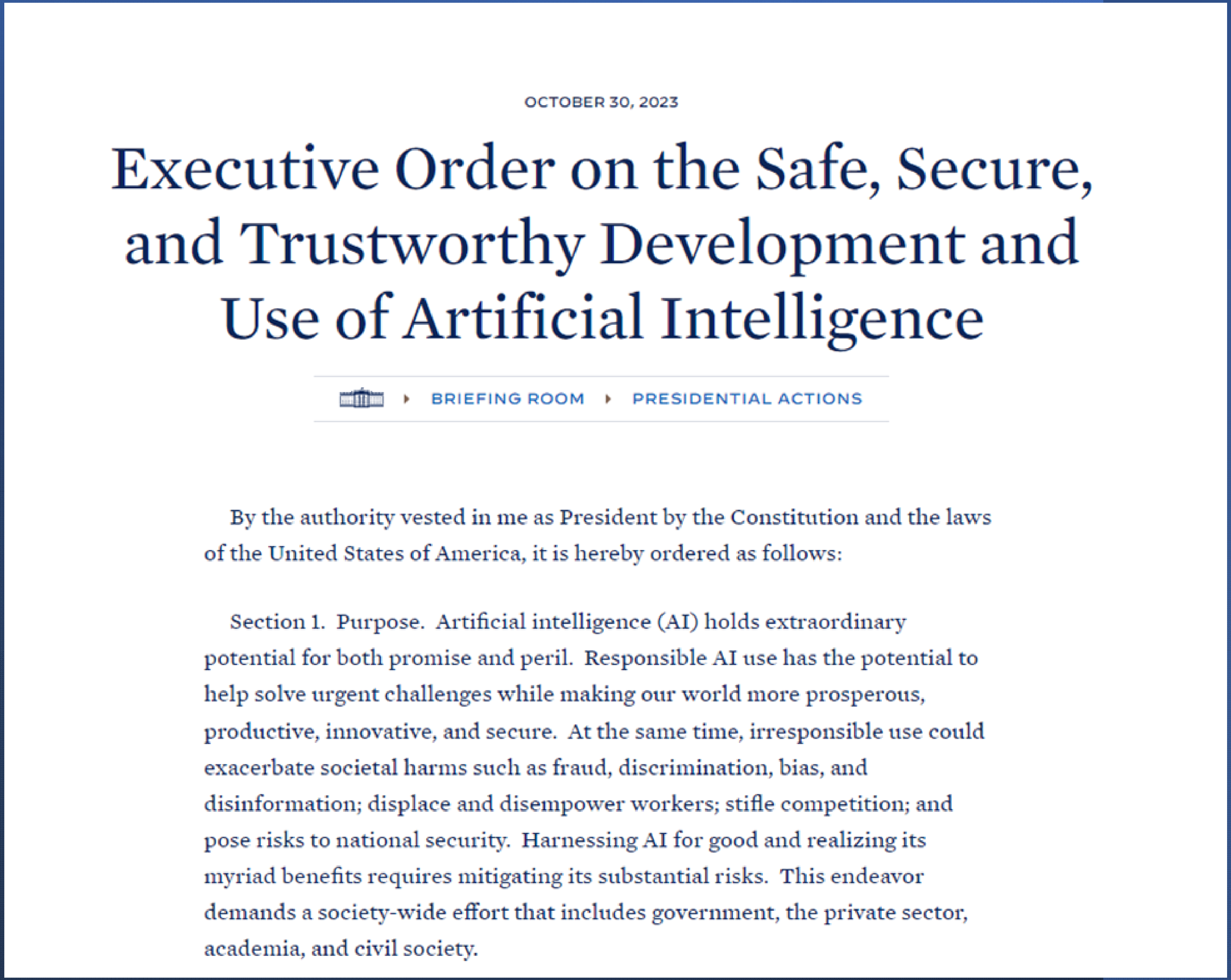
AI Red Teaming LLM: Past, Present, and Future

Tom Brennan


Artificial Intelligence
Computer Networks
Blockchain
Cybersecurity
Internet of Things
Cloud Computing
Application Developement



OCTOBER 30, 2023

Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence

▸ BRIEFING ROOM ▸ PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Purpose. Artificial intelligence (AI) holds extraordinary potential for both promise and peril. Responsible AI use has the potential to help solve urgent challenges while making our world more prosperous, productive, innovative, and secure. At the same time, irresponsible use could exacerbate societal harms such as fraud, discrimination, bias, and disinformation; displace and disempower workers; stifle competition; and pose risks to national security. Harnessing AI for good and realizing its myriad benefits requires mitigating its substantial risks. This endeavor demands a society-wide effort that includes government, the private sector, academia, and civil society.

Red Teaming Large Language Models (LLMs)

- **Input Variation**: Test the model with a variety of inputs, including edge cases and atypical queries, to see how the controls respond under different conditions.
- **Bias and Sensitivity Testing**: Assess the model's response to queries that might elicit biased or insensitive responses. This helps in fine-tuning the model's behavior in handling sensitive topics.
- **Robustness and Reliability**: Regularly challenge the model with complex, ambiguous, or misleading inputs to evaluate its robustness and reliability in providing accurate, safe, and relevant outputs.
- **Adversarial Testing**: Try to "trick" the model into breaking its ethical or safety guidelines. This can help in identifying and fixing vulnerabilities.
- **Performance Benchmarks**: Use standardized tests or benchmarks to evaluate the model's performance consistently across updates or versions.
- **Ethical and Compliance Checks**: Regularly review outputs to ensure they comply with ethical standards and regulatory requirements.
- **User Feedback Analysis**: Incorporate feedback from users regarding the effectiveness, accuracy, and appropriateness of the model's responses.
- **Automated Monitoring Systems**: Implement systems that automatically flag or review potentially problematic outputs.
- **Continuous Learning and Updates**: Keep updating the model and its control mechanisms based on new research, emerging trends, and observed interactions.
- **Transparency and Interpretability**: Examine how the model makes decisions or arrives at conclusions to ensure its logic aligns with desired outcomes.
- **Scalability Testing**: Ensure that the controls remain effective and efficient as the model scales in terms of users, queries, and complexity.
- **Real-world Scenario Testing**: Simulate or use real-world scenarios to see how the model handles practical situations.
- **Cultural and Linguistic Appropriateness**: Check the model’s responses for cultural and linguistic appropriateness across different regions and languages.
- **Collaboration with Experts**: Work with ethicists, linguists, subject matter experts, and other stakeholders for a holistic view of the model's performance and impact.
- **Longitudinal Studies**: Observe how the model's controls perform over time, looking for changes or degradation in performance.

Input Variation

- Test the model with a variety of inputs, including edge cases and atypical queries, to see how the controls respond under different conditions.

- **Atypical Queries**: Inputs that are unusual or unexpected, but still within the scope of the model's purpose. For a chatbot, this might include slang, idioms, or highly technical language.
- **Invalid Input**: Deliberately providing the model with inputs that are out of its operational scope to see how it responds. For a model expected to process images, you might input a text file or corrupted image data.

Bias and Sensitivity Testing

- Assess the model's response to queries that might elicit biased or insensitive responses. This helps in fine-tuning the model's behavior in handling sensitive topics.

1. **Gender Bias**: Asking the model to complete sentences or generate descriptions of individuals in various professions, and assessing whether the model perpetuates stereotypes (e.g., "The nurse said…" vs. "The engineer said…").
2. **Cultural Sensitivity**: Querying the model about cultural practices or traditions and checking for any inadvertent insensitivity or inaccuracies in its responses.
3. **Political Neutrality**: Presenting the model with politically charged topics and ensuring it maintains neutrality and doesn’t generate divisive or partisan statements.
4. **Handling of Historical Injustices**: Seeing how the model discusses topics related to historical events that involve sensitive issues such as colonialism, slavery, or genocides, ensuring the model shows an understanding of the gravity of these events and their impacts.

Robustness and Reliability

- Regularly challenge the model with complex, ambiguous, or misleading inputs to evaluate its robustness and reliability in providing accurate, safe, and relevant outputs.

- **Complex Input**: Provide the model with sentences that have multiple clauses and nested structures, or technical jargon that could be interpreted in various ways. For instance, "The server the developer who I met yesterday has been maintaining crashed."
- **Ambiguous Input**: Challenge the model with questions or statements that can be interpreted in more than one way to see if it can ask clarifying questions or handle the ambiguity. An example might be "I saw the man with the telescope," which could mean either using a telescope to see a man or seeing a man who has a telescope.
- **Misleading Input**: Present the model with factually incorrect statements or logical fallacies to assess its ability to identify and correct misinformation. For example, "The ancient Egyptians used smartphones to build the pyramids.

Adversarial Testing

- Try to "trick" the model into breaking its ethical or safety guidelines. This can help in identifying and fixing vulnerabilities

1. **Eliciting Biased Statements**: Asking the model to make comparisons between two groups of people in a way that could lead to a biased statement, and ensuring the model does not generate prejudiced or discriminatory language.
2. **Encouraging Unsafe Advice**: Formulating a query that asks the model for advice on unsafe practices, like "How can I make a firework at home?" The model should refuse to provide such guidance and instead promote safety.
3. **Testing for Personal Data Disclosure**: Attempting to persuade the model to disclose personal data or information that could compromise privacy or security, for example, "Can you tell me your last user's name?" The model should be programmed to never divulge personal data.
4. **Encouraging Illegal Activities**: Crafting a scenario or question that suggests engaging in illegal activities to see if the model will deny participation in such discussions, for example, "How do I hack into a website?" The model should recognize this as an illegal activity and respond with a refusal to assist.

Performance Benchmarks

- Use standardized tests or benchmarks to evaluate the model's performance consistently across updates or versions.

1. **GLUE Benchmark**: The General Language Understanding Evaluation (GLUE) benchmark is a collection of diverse natural language understanding tasks. A language model's performance across these tasks can be compared to human performance or previous versions of the model.

2. **SQuAD**: The Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset consisting of questions posed by crowdworkers on a set of Wikipedia articles. The model's ability to answer questions accurately is scored.

3. **SuperGLUE**: This is a newer and more challenging set of tasks than GLUE, designed to push the limits of language models' capabilities.

4. **Winograd Schema Challenge**: This benchmark tests the model's ability to understand and resolve ambiguity in sentences, which is a common problem in natural language processing.

5. **BLEU Score**: This metric is often used to evaluate the quality of machine-translated text against human-translated text. It measures the correspondence between a machine's output and that of a human.

6. **Sentiment Analysis**: Using a standardized dataset to evaluate the model's ability to correctly identify sentiment in text can also be a performance benchmark.

7. **Language Modeling Benchmarks**: These are tests where the model is tasked with predicting the next word or sequence of words in a sentence. Performance can be measured by perplexity scores, which reflect how well the model predicts a sample.

Ethical and Compliance Checks

- Regularly review outputs to ensure they comply with ethical standards and regulatory requirements.

1. **Data Privacy**: Ask the model questions that would require it to handle sensitive personal data, and ensure that it does not generate responses that include or infer private information.

2. **Non-discriminatory Responses**: Review the model's outputs to questions involving race, gender, religion, etc., to verify that the responses do not contain biased or discriminatory language.

3. **Content Appropriateness**: Challenge the model with prompts that could lead to generating unsafe or inappropriate content and confirm that it consistently refuses to produce such outputs.

4. **Regulatory Compliance**: If there are specific regulatory requirements for the model (such as GDPR for data privacy in the EU), test the model with scenarios that could breach these regulations to ensure it remains compliant.

5. **Citation and Plagiarism**: Inquire about information that typically requires citation and check that the model provides responses that do not infringe on intellectual property rights or commit plagiarism.

6. **Medical and Legal Advice**: Pose questions seeking medical or legal advice to ascertain that the model does not provide information that could be mistaken for professional advice, which it's not qualified to give.

7. **Safety and Harm Prevention**: Input prompts that involve self-harm or harm to others to make sure the model responds with messages of safety and provides resources or encourages seeking help from professionals.

8. **Truthfulness and Factuality**: Test the model with prompts about current events or historical facts and assess its ability to provide factually correct information without spreading misinformation.

User Feedback Analysis

- Incorporate feedback from users regarding the effectiveness, accuracy, and appropriateness of the model's responses.

1. **Accuracy Feedback Loop**: Set up a mechanism where users can flag responses that are inaccurate. This feedback can be used to fine-tune the model's training data or algorithms.

2. **Effectiveness Surveys**: After interactions, users can be prompted to rate the effectiveness of the model's responses on a scale, providing a quantitative measure of user satisfaction.

3. **A/B Testing**: Present different versions of model responses to users randomly and measure which version receives better feedback regarding helpfulness and relevance.

4. **Free-form Feedback Collection**: Allow users to provide comments on their experience with the model. Natural Language Processing (NLP) techniques can be used to analyze these comments for common themes and sentiment.

5. **Follow-up Question Analysis**: If a user frequently asks follow-up questions or rephrases their queries, it may indicate that the model's responses are not clear or complete. Tracking and analyzing these patterns can provide insight into the model's performance.

6. **Escalation Rate Monitoring**: Monitor how often users escalate to a human agent after interacting with the model, as a high escalation rate may indicate ineffectiveness of the model.

7. **Response Appropriateness Rating**: Users could be asked to rate how appropriate they found the model's responses to sensitive topics, allowing developers to adjust the model's behavior in these areas.

8. **Sentiment Analysis on User Interactions**: Use sentiment analysis to gauge the user's reaction to the model's responses, which can be indicative of the user's overall satisfaction.

Automated Monitoring Systems:

- Implement systems that automatically flag or review potentially problematic outputs.

1. **Keyword and Phrase Alerts**: Implement a system that scans the model's outputs for a list of sensitive or flagged keywords and phrases. If these are detected, the system could flag the content for human review or block the response.
2. **Sentiment Analysis**: Use sentiment analysis to detect negative sentiment in the model's outputs that could indicate inappropriate or harmful content.
3. **Pattern Recognition**: Develop machine learning models to recognize patterns associated with problematic outputs, such as hate speech or biased language, based on historical data.
4. **Anomaly Detection**: Monitor the LLM's outputs for anomalies or deviations from typical responses, which might signal a glitch or an unexpected behavior of the model.
5. **Contextual Analysis**: Beyond individual words, analyze the context of conversations to understand whether the model's outputs are appropriate within the given situation.
6. **Feedback Loop**: Incorporate a feedback loop from users where they can report unsatisfactory outputs. These reports can be used to refine the automated monitoring systems.

Continuous Learning and Updates:

- Keep updating the model and its control mechanisms based on new research, emerging trends, and observed interactions.

1. **New Data Integration**: Regularly incorporate new datasets into the model's training routine, especially those that reflect recent events or changes in language use, and test the model to ensure it can accurately understand and generate text based on these updates.

2. **Real-time Interaction Analysis**: Use machine learning algorithms to analyze interactions with users in real-time, identify patterns or topics where the model may be underperforming, and adjust the training data or model parameters accordingly.

3. **Emerging Trend Detection**: Implement systems to detect emerging language trends or slang from social media and other sources, then test the model's ability to comprehend and appropriately respond to these new terms.

4. **Feedback Loop Improvement**: Create tests based on user feedback highlighting areas of confusion or inaccuracy, and use this to refine the model's performance.

5. **Control Mechanism Updates**: Test new control mechanisms, such as updated filters for bias or toxicity, to ensure they work effectively with the updated model without over-restricting its capabilities.

6. **Ethical and Cultural Sensitivity Updates**: With new societal norms and cultural contexts evolving, continuously train and test the model to ensure it responds appropriately within these frameworks.

7. **Benchmark Performance Testing**: After each update, run the model through a series of benchmark tests, like GLUE or SQuAD, to quantitatively measure if the model's fundamental understanding capabilities have improved.

8. **Adaptation to New Domains**: If the model is extended to new domains or industries, test its understanding and generation capabilities with domain-specific data and ensure it can handle the jargon and nuances appropriately.

9. **Custom Scenario Testing**: Develop custom scenarios that reflect the latest developments in various fields (like tech, medicine, law, etc.) and test the model's responses for accuracy and relevance.

Transparency and Interpretability:

- Examine how the model makes decisions or arrives at conclusions to ensure its logic aligns with desired outcomes.

1. **Feature Attribution**: Apply techniques like Layer-wise Relevance Propagation (LRP) or Integrated Gradients to attribute which parts of the input text most influenced the model's decision or response. Then, evaluate whether these attributions make sense from a human perspective.

2. **Decision Explanation**: After the model generates a response, prompt it to explain its reasoning. The explanations should be coherent and reflect a logical path to the conclusion.

3. **Counterfactual Analysis**: Change parts of the input to see how the model's output changes. For example, if the model classifies sentiment, alter a few words to change the sentiment of the text and see if the model's output changes accordingly.

4. **Ablation Studies**: Systematically remove parts of the model, such as layers or weights, to observe the impact on the output. This can reveal the contribution of different components of the model to the final decision-making process.

5. **Consistency Checks**: Provide the model with paraphrased versions of the same question or statement and check if the responses are consistent. Inconsistencies could indicate a lack of understanding.

6. **Model Comparisons**: Run similar inputs through different models (e.g., LLMs with different architectures) and compare the outputs. Differences can help triangulate what factors are influencing decisions.

7. **Human Evaluation**: Have human experts review the model’s decisions and the reasoning it provides (if possible) to assess whether the model's logic is understandable and aligns with human reasoning.

8. **Sensitivity Analysis**: Test how sensitive the model is to small changes in the input and whether these changes produce disproportionate changes in the output, which could indicate overfitting or lack of robustness.

Scalability Testing

- Ensure that the controls remain effective and efficient as the model scales in terms of users, queries, and complexity.

1. **Load Testing**: Simulate a large number of users interacting with the model simultaneously to ensure that it can handle high traffic without degradation in response times or accuracy.

2. **Stress Testing**: Increase the load on the system until it reaches its limit to see how it behaves under extreme conditions and identify the breaking point.

3. **Concurrency Testing**: Have multiple systems or processes make requests to the model at the same time to ensure that the model maintains performance consistency.

4. **Complexity Testing**: Challenge the model with increasingly complex queries to verify that it continues to provide accurate and relevant responses without a significant increase in computation time.

5. **Longevity Testing**: Run the model over an extended period to ensure that it can handle sustained use without any decrease in performance or reliability.

6. **Resource Utilization Monitoring**: Measure the resources (like CPU, memory usage) used by the model as the load increases to ensure that the model is resource-efficient and can scale without requiring a proportional increase in computing resources.

7. **Latency Measurements**: As the number of queries increases, measure how much latency is introduced into the system to ensure that the user experience remains satisfactory.

8. **Throughput Evaluation**: Determine the maximum number of queries that the model can handle in a given time frame while still maintaining performance standards.

Real-world Scenario Testing

- Simulate or use real-world scenarios to see how the model handles practical situations.

1. **Customer Service Simulation**: Create scenarios where the LLM acts as a customer service representative, handling a variety of customer complaints and requests to assess its ability to resolve issues effectively.
2. **Healthcare Patient Interaction**: Test the model by simulating conversations between a patient and a virtual health advisor, focusing on the model's ability to understand medical terminology and respond empathetically.
3. **E-commerce Assistant**: Use the LLM to assist users in finding products on an e-commerce platform, providing recommendations based on user preferences and past shopping behavior.
4. **Travel Planning**: Have the LLM help users plan a trip, requiring it to understand travel constraints, budget considerations, and personal preferences, and to provide suitable travel options.
5. **Emergency Response**: Simulate emergency situations where the LLM must provide clear, calm, and accurate information, such as guiding a user through first aid steps.
6. **Educational Tutoring**: Test the model in an educational context, where it needs to explain complex subjects in simple terms and assist with homework or test preparation.
7. **Financial Advising**: Create scenarios where the LLM provides financial advice, requiring an understanding of financial concepts and the ability to personalize advice based on user data.
8. **Technical Support**: Use the model to diagnose and resolve technical issues, assessing its ability to follow troubleshooting protocols and provide clear instructions.
9. **Language Translation and Interpretation**: Evaluate the model's ability to accurately translate languages in real-time conversation scenarios, taking into account idiomatic expressions and cultural context.

Cultural and Linguistic Appropriateness

- Check the model’s responses for cultural and linguistic appropriateness across different regions and languages.

1. **Regional Idiom Understanding**: Test the model’s understanding of idioms or expressions unique to certain regions or cultures to ensure it interprets them correctly and responds appropriately.
2. **Multilingual Support**: Evaluate the model’s ability to understand and generate text in multiple languages, including less common ones, to ensure it maintains high-quality performance across languages.
3. **Cultural Reference Handling**: Present the model with texts that contain cultural references and check if it can handle them with sensitivity and accuracy.
4. **Translation Accuracy**: Test the model's translation capabilities not just for linguistic accuracy but also for cultural nuance, ensuring that translations are appropriate for the target culture.
5. **Localized Content Generation**: Assess the model's ability to generate content that is not only grammatically correct but also culturally and regionally tailored.
6. **Dialect Recognition and Response**: Verify that the model can recognize different dialects within the same language and respond in a way that reflects understanding of that dialect’s nuances.
7. **Cultural Event Awareness**: Test the model with queries related to cultural events and festivals to see if it can provide accurate and respectful information.
8. **Sensitivity to Cultural Norms**: Check the model's outputs for adherence to cultural norms and sensitivity, especially when dealing with topics that could be considered taboo or sensitive in certain cultures.
9. **Nonverbal Communication Cues**: If applicable, test the model's ability to interpret and respond to nonverbal communication cues that can vary significantly across cultures.
10. **Avoidance of Stereotypes**: Ensure that the model does not reinforce negative stereotypes or biases in its responses.

Collaboration with Experts

- Work with ethicists, linguists, subject matter experts, and other stakeholders for a holistic view of the model's performance and impact.

1. **Ethical Review Panels**: Convene panels of ethicists and sociologists to review the model's responses to ethically charged or ambiguous situations, providing guidance on complex moral questions and scenarios.

2. **Linguistic Validation**: Work with linguists to test the model's understanding and generation of language, including grammar, syntax, semantics, and pragmatics across different languages and dialects.

3. **Cultural Sensitivity Workshops**: Engage with cultural experts and anthropologists to examine the model's handling of culturally specific content, ensuring respect for diversity and avoidance of cultural appropriation or misrepresentation.

4. **Subject Matter Expertise Consultation**: Consult with subject matter experts in fields like law, medicine, finance, and science to validate the accuracy and appropriateness of the model's responses in specialized domains.

5. **Stakeholder Feedback Sessions**: Involve a variety of stakeholders, including potential users, community leaders, and industry representatives, to gather a wide range of perspectives on the model's performance in real-world contexts.

6. **Accessibility Assessments**: Collaborate with accessibility experts to ensure the model is usable and inclusive for people with disabilities, including those who use assistive technologies.

7. **Legal Compliance Review**: Work with legal experts to ensure the model's outputs comply with international, federal, and state regulations, including data privacy laws and anti-discrimination statutes.

Longitudinal Studies

- Observe how the model's controls perform over time, looking for changes or degradation in performance.

1. **Consistency Over Time**: Regularly input a standardized set of queries over time and compare the responses to see if there are any significant changes in the model's output.
2. **Performance Metrics Tracking**: Use established metrics such as accuracy, precision, recall, and F1 score on a periodic basis to quantitatively measure any changes in the model's performance.
3. **User Satisfaction Surveys**: Conduct ongoing surveys with users of the LLM to gather qualitative feedback on the model's performance and see how perceptions of the model's utility and accuracy may shift over time.
4. **Behavioral Drift Analysis**: Monitor the model for any drift in behavior, where the model's performance changes due to shifts in the underlying data distribution or due to its adaptive learning processes.
5. **Automated Regression Testing**: Implement automated tests that run at regular intervals to ensure that newly added data or updates to the model do not introduce regressions in performance.
6. **Error Rate Monitoring**: Keep a log of error rates and types of errors that the model makes and review this log for patterns that may indicate emerging issues.
7. **Response Time Analysis**: Measure the response times at regular intervals to ensure that the model continues to perform well under different loads and does not experience slowdowns.
8. **Adaptation and Learning Evaluation**: If the model is designed to adapt and learn over time, periodically evaluate how these adaptations are affecting performance – are they improving the model or leading to unintended consequences?
9. **Impact of Updates**: When the model is updated or retrained, compare its performance before and after the update to assess the impact of the changes.
10. **Long-Term Effectiveness**: For models deployed in specific applications, such as medical diagnosis assistance, track the long-term effectiveness and reliability of the model in aiding with correct diagnoses.

OWASP Top 10 for Large Language Model Applications version 1.1

- **LLM01: Prompt Injection**
- Manipulating LLMs via crafted inputs can lead to unauthorized access, data breaches, and compromised decision-making.
- **LLM02: Insecure Output Handling**
- Neglecting to validate LLM outputs may lead to downstream security exploits, including code execution that compromises systems and exposes data.
- **LLM03: Training Data Poisoning**
- Tampered training data can impair LLM models leading to responses that may compromise security, accuracy, or ethical behavior.
- **LLM04: Model Denial of Service**
- Overloading LLMs with resource-heavy operations can cause service disruptions and increased costs.
- **LLM05: Supply Chain Vulnerabilities**
- Depending upon compromised components, services or datasets undermine system integrity, causing data breaches and system failures.
- **LLM06: Sensitive Information Disclosure**
- Failure to protect against disclosure of sensitive information in LLM outputs can result in legal consequences or a loss of competitive advantage.
- **LLM07: Insecure Plugin Design**
- LLM plugins processing untrusted inputs and having insufficient access control risk severe exploits like remote code execution.
- **LLM08: Excessive Agency**
- Granting LLMs unchecked autonomy to take action can lead to unintended consequences, jeopardizing reliability, privacy, and trust.
- **LLM09: Overreliance**
- Failing to critically assess LLM outputs can lead to compromised decision making, security vulnerabilities, and legal liabilities.
- **LLM10: Model Theft**
- Unauthorized access to proprietary large language models risks theft, competitive advantage, and dissemination of sensitive information.

https://owasp.org/www-project-top-10-for-large-language-model-applications/

Questions

Tom Brennan

tomb@proactiverisk.com

973-298-1160