Documento Técnico Complementar
Técnicas de Engenharia de Prompt: PHP, HTP e SHP

Alex Sandre Pinheiro Severo[1], **Douglas Paim Lautert**[1], **Gefte Almeida**[1]
Diego Kreutz[1], **Godinho Rodrigo**[2], **Lourenco A. Pereira Jr**[3], **Leandro M. Bertholdo**[4]

[1] Universidade Federal do Pampa (UNIPAMPA)

[2]VALE

[3]Instituto Tecnológico de Aeronáutica (ITA)

[4] Universidade Federal do Rio Grande do Sul (UFRGS)

Resumo. *Este documento apresenta três técnicas de engenharia de prompt aplicadas à tarefa de categorização automatizada de incidentes de segurança da informação: Progressive-Hint Prompting (PHP), Hypothesis Testing Prompting (HTP) e Self-Hint Prompting (SHP). As técnicas foram adaptadas para operar sobre descrições textuais fornecidas por CSIRTs e SOCs, utilizando modelos de linguagem de grande porte (LLMs) e tendo como referência a taxonomia do NIST SP 800-61r3. O PHP adota um mecanismo iterativo de refinamento por meio de dicas geradas a partir das saídas anteriores do próprio modelo. O HTP estrutura o raciocínio como um processo de testagem contrastiva de hipóteses. O SHP, por sua vez, induz ciclos de autorreflexão para detecção e correção de falhas inferenciais. As três abordagens foram selecionadas por sua independência metodológica e complementaridade cognitiva, visando à robustez semântica e à confiabilidade classificatória em ambientes narrativos e ambíguos.*

1. Introdução

Este documento técnico apresenta, de forma complementar ao artigo principal, três técnicas de engenharia de prompt utilizadas no processo de categorização automatizada de incidentes de segurança: ***Progressive-Hint Prompting*** **(PHP)**, ***Hypothesis Testing Prompting*** **(HTP)** e ***Self-Hint Prompting*** **(SHP)**. O objetivo é detalhar a origem, finalidade, adaptação e aplicação prática dessas abordagens, bem como apresentar pseudocódigos e exemplos da implementação realizada neste estudo. Para aplicações sobre LLMs ou SLMs foram utilizados as dependências relacionadas ao Huggingface e OpenAI como base para SLMs e LLMs.

2. Técnica PHP – Progressive-Hint Prompting

Origem

A técnica *Progressive-Hint Prompting* (PHP) foi proposta por [Zheng et al. 2023] como uma abordagem de engenharia de prompt destinada a aprimorar o desempenho de raciocínio matemático em *Large Language Models* (LLMs). O método parte da observação de que os LLMs muitas vezes falham em tarefas complexas por não conseguirem manter coerência argumentativa ou aprofundar inferências em um único ciclo de geração textual.

A ideia central do PHP é introduzir um processo iterativo de raciocínio, no qual o modelo é encorajado a reconsiderar suas respostas anteriores à luz de "dicas" (*hints*) geradas progressivamente. Essas dicas são compostas pelas próprias saídas anteriores do modelo, realimentadas como contexto em rodadas subsequentes de inferência.

Funcionamento do PHP

O funcionamento da técnica pode ser descrito em três etapas:

1. **Geração Inicial** (Q_0): o modelo recebe uma pergunta e um prompt inicial (por exemplo, estilo *Chain-of-Thought*) e gera uma primeira resposta A_1.
2. **Iterações com Dica**: se a resposta não for satisfatória (com base em critérios objetivos ou heurísticos), constrói-se um novo *prompt* contendo uma ou mais das respostas anteriores como DICAS. Esse *prompt* reformulado é enviado novamente ao modelo para gerar a resposta A_2, e assim sucessivamente.
3. **Critério de Parada**: o processo se encerra quando duas respostas consecutivas forem semanticamente equivalentes (por exemplo, com ROUGE-L $\geq 0{,}9$), ou quando um número máximo de iterações for atingido.

Exemplo Ilustrativo

No domínio de raciocínio matemático, a pergunta:

"Quantos lados tem a soma de três triângulos?"

poderia inicialmente levar a uma resposta incorreta (e.g., *6*). Na rodada seguinte, o prompt é enriquecido com a dica:

"Sabemos que uma dica é que cada triângulo tem 3 lados."

Com essa informação adicional, o modelo é conduzido à resposta correta: *9*.

Justificativa Cognitiva

O PHP simula um processo de autoverificação análogo ao raciocínio humano, no qual hipóteses anteriores são reconsideradas com base em novas evidências ou revisões. Isso torna o modelo menos propenso a erros factuais ou conclusões precipitadas.

Potencial de Generalização

Embora o foco original da técnica tenha sido o raciocínio matemático, os autores indicam que o PHP pode ser adaptado para diversas tarefas que exigem inferência complexa, tais como:

- Resolução de problemas lógico-matemáticos;
- Diagnóstico clínico automatizado;
- Análise jurídica e tomada de decisão;
- Tradução contextual;

A adaptabilidade da técnica a outros domínios depende da definição apropriada de critérios de parada e da construção eficaz de pistas intermediárias.

Aplicação do PHP na Classificação de Incidentes de Segurança

A técnica *Progressive-Hint Prompting* (PHP), originalmente proposta por [Zheng et al. 2023] para melhorar o raciocínio matemático em LLMs, foi adaptada neste trabalho para a tarefa de classificação automatizada de incidentes de segurança cibernética, com base em relatos reais fornecidos por SOCs e CSIRTs. A principal ideia da adaptação consiste em transformar o processo de raciocínio iterativo em um mecanismo de refinamento progressivo de categorização, utilizando como referência a taxonomia estabelecida pelo NIST SP 800-61r3.

Nesta adaptação, cada relato textual de incidente é processado por um LLM a partir de um *prompt base*. A resposta inicial é tratada como uma primeira hipótese de categorização, a partir da qual extrai-se uma "*hint*' (dica), geralmente a categoria sugerida, como `CAT2` ou `CAT4`. Essa dica é então incorporada a um novo prompt, refinando a consulta ao modelo. Esse processo é repetido iterativamente até que duas respostas consecutivas sejam semanticamente similares, com base em uma métrica de similaridade textual.

Para mensuração da convergência, são utilizadas técnicas de avaliação automática, como a métrica ROUGE-L e a similaridade de *embeddings* obtida por modelos como MiniLM. A convergência semântica é atingida quando a similaridade entre as respostas atinge um limiar pré-definido (e.g., 0,9). Ao final do processo, a classificação é considerada estabilizada e a categoria mais provável é atribuída ao incidente.

Diferentemente da aplicação original em problemas matemáticos, onde o objetivo era corrigir erros de cálculo, a versão aplicada à cibersegurança busca alinhar a narrativa textual do incidente com descrições semânticas associadas às categorias da taxonomia NIST. As dicas são geradas automaticamente com base nas saídas do próprio modelo, o que evita a necessidade de intervenção humana contínua durante o processo.

Esta abordagem demonstrou resultados expressivos quando combinada com o modelo Gemini 2, alcançando 92,27% de correspondência com as classificações realizadas por especialistas humanos. Além da acurácia, o PHP também se destacou em termos de eficiência temporal e uso otimizado de tokens, sendo mais econômico que outras estratégias como SHP e HTP. Assim, sua aplicação prática é recomendada em cenários que exigem alta precisão e baixa latência operacional na triagem de incidentes.

Pseudocódigo PHP

Algorithm 1: Progressive-Hint Prompting (PHP)

Input: Prompt I contendo a descrição do incidente
Output: Resposta final estável Y_i

$Y_i \leftarrow$ geraResposta(I);
$i \leftarrow 0$;
$hintMax \leftarrow 4$;
$limiar \leftarrow 0.9$;
repeat
  $H_i \leftarrow$ geraDica(Y_i);
  $Y_i \leftarrow$ geraResposta(H_i);
  $X_i \leftarrow Y_i$;
  $i \leftarrow i + 1$;
until *similaridade*(Y_i, X_i) $\geq limiar$ ***ou*** $i \geq hintMax$;
return Y_i

3. Técnica HTP – Hypothesis Testing Prompting

Origem

A técnica *Hypothesis Testing Prompting* (HTP) foi proposta por [Li et al. 2024] com o objetivo de aprimorar o desempenho de modelos de linguagem em tarefas de raciocínio dedutivo complexo, superando limitações observadas em abordagens como *Chain-of-Thought*. Inspirada no método científico e nos procedimentos formais de testes de hipóteses estatísticas, a HTP foi avaliada nos *benchmarks* RuleTaker e ProofWriter, demonstrando ganhos significativos na precisão e na coerência dos processos inferenciais intermediários.

Finalidade

A HTP estrutura o raciocínio como um processo investigativo, em que o modelo assume inicialmente que uma hipótese é verdadeira e conduz uma sequência de inferências baseadas em regras e fatos fornecidos. Em seguida, a hipótese é assumida como falsa e o mesmo processo é repetido. A partir dessa análise reversa e contrastiva, o modelo é capaz de:

- Confirmar a hipótese (caso as inferências a sustentem);
- Refutá-la (caso as inferências levem a contradições);
- Ou classificá-la como "desconhecida"(quando nenhuma das suposições pode ser sustentada logicamente).

Essa abordagem leva o modelo a realizar validações cruzadas de suas próprias conclusões, promovendo maior robustez e precisão em cenários onde múltiplas interpretações podem coexistir. Além disso, a HTP demonstrou desempenho superior na identificação de respostas classificadas como "*unknown*" — um ponto fraco recorrente em técnicas mais simples como CoT.

Aplicação em Raciocínio Dedutivo

A técnica mostrou especial eficácia em tarefas que envolvem cadeias lógicas de múltiplos passos, como aquelas exigidas em provas matemáticas, lógicas formais e sistemas baseados em regras. Ao seguir um fluxo que espelha a prática científica (afirmação, teste, refutação), a HTP não apenas aumenta a acurácia das respostas, mas também torna os processos inferenciais mais interpretáveis e auditáveis.

Aplicação do HTP na Classificação de Incidentes de Segurança

A técnica *Hypothesis Testing Prompting* (HTP), originalmente voltada ao raciocínio dedutivo, foi adaptada neste estudo para a tarefa de categorização automatizada de incidentes de segurança da informação. Sua aplicação nesse domínio visa confrontar suposições categóricas — representadas por rótulos como `CAT1`, `CAT4`, etc. — com os fatos extraídos das descrições textuais dos incidentes, baseando-se na lógica de teste de hipóteses.

O processo parte da formulação de uma hipótese inicial sobre a categoria provável do incidente com base na taxonomia NIST SP 800-61r3. Essa hipótese é então submetida ao modelo de linguagem com dois caminhos de verificação: uma suposição de que a

hipótese é verdadeira e uma suposição de que é falsa. A partir dessas formulações contrastivas, o modelo é conduzido a avaliar se a hipótese é logicamente sustentada pelos fatos fornecidos na narrativa do incidente.

Os prompts foram estruturados de forma a explicitar esse raciocínio condicional, guiando o modelo a inferir relações lógicas entre a hipótese proposta e os elementos do relato. Por exemplo, para um incidente envolvendo exfiltração de dados, o prompt pode incluir: "Suponha que o incidente seja da categoria CAT4 (Vazamento de Dados). Quais fatos do texto sustentam ou refutam essa hipótese?"

Essa abordagem exploratória permite que o modelo simule um raciocínio bifurcado, promovendo a validação da hipótese com base em inferência textual. Caso os argumentos derivados da hipótese se alinhem com os fatos extraídos, a categoria é mantida. Caso contrário, a hipótese é rejeitada, e uma nova hipótese pode ser testada.

Diferente de técnicas como o *Progressive-Hint Prompting*, que refinam iterativamente a saída, a HTP promove uma avaliação crítica da suposição inicial por meio de análise lógica estruturada. Essa propriedade torna a HTP especialmente útil em casos ambíguos, nos quais múltiplas categorias podem ser plausíveis. Ao induzir o modelo a pensar sob condições contrastantes, a técnica melhora a robustez semântica da classificação e reduz o viés confirmatório comum em técnicas de prompt diretas.

A adoção do HTP na categorização de incidentes demonstrou complementaridade com heurísticas como PHP, principalmente em cenários onde a validação lógica da categoria era essencial para evitar classificações imprecisas ou genéricas. Como resultado, essa técnica contribui significativamente para o aprimoramento da consistência e da justificativa semântica das decisões classificatórias tomadas pelas LLMs.

Pseudocódigo HTP

Algorithm 2: Hypothesis Testing Prompting (HTP)

Input: Incidente I, Lista de categorias $CATEGORIES = [CAT1, CAT2, ..., CAT12]$, Palavras-chave por categoria, Limiar de similaridade $LIMIAR$
Output: Categoria classificada e justificativa

$i \leftarrow 0$;
while $i < |CATEGORIES|$ **do**
  $CATEGORIA \leftarrow CATEGORIES[i]$;
  $KEYWORDS \leftarrow obterPalavrasChave(CATEGORIA)$;
  `// Gerar prompt de testagem de hipóteses`
  $PROMPT \leftarrow$ gerarPromptHTP($I, CATEGORIA, KEYWORDS$);
  $HIPOTESES \leftarrow enviar\_para\_LLM(PROMPT)$;
  `// Analisar resposta com base nas hipóteses`
  if *HIPÓTESE verdadeira é SUPORTADA e HIPÓTESE falsa é NÃO SUPORTADA* **then**
    $categoria \leftarrow CATEGORIA$;
    $explicacao \leftarrow$ justificativa extraída de $HIPOTESES$;
  else
    $categoria \leftarrow UNKNOWN$;
    $explicacao \leftarrow UNKNOWN$;
  end
  `// Verificar similaridade semântica`
  if $similaridade(categoria, CATEGORIA) \geq LIMIAR$ **then**
    break;
  end
  $i \leftarrow i + 1$;
end
return *categoria, explicacao*

4. Técnica SHP – Self-Hint Prompting

Origem

Proposta por [Chen et al. 2024], a técnica *Self-Hint Prompting* (SHP) baseia-se no ciclo reflexivo de Kolb (2014), que descreve a aprendizagem como um processo cíclico de experiência concreta, observação reflexiva, conceitualização abstrata e experimentação ativa. SHP simula esse processo em modelos de linguagem, permitindo que eles revisem seus próprios planos de raciocínio e ajustem suas estratégias com base em erros de compreensão ou de inferência lógica detectados em etapas anteriores.

Finalidade

O objetivo do SHP é estimular a autorreflexão nos LLMs, promovendo ajustes iterativos em seus planos de resolução. A técnica introduz um mecanismo de reflexão consciente, no qual o modelo é guiado a detectar e corrigir falhas de raciocínio ou compreensão a partir da análise de suas próprias respostas intermediárias.

A SHP é particularmente eficaz em ambientes de raciocínio de múltiplos passos, como problemas matemáticos ou dedutivos, nos quais a falha em um único passo pode comprometer toda a cadeia de inferência. Com SHP, o modelo é levado a:

- Revisar criticamente os planos intermediários gerados;
- Identificar erros de compreensão semântica (e.g., interpretação incorreta de conceitos);
- Corrigir falhas no raciocínio (e.g., passos inválidos, omissões ou desvios);
- Reexecutar a tarefa com base em um plano ajustado.

Funcionamento Geral

A técnica segue os seguintes estágios, inspirados diretamente no ciclo de Kolb:

1. **Experiência concreta:** o LLM gera um plano inicial e resolve o problema com base em seu conhecimento prévio e no prompt de entrada.
2. **Observação objetiva:** o plano e a resposta são capturados como observações externas.
3. **Reflexão consciente:** o modelo é instruído a examinar a qualidade do plano e detectar falhas, com foco em problemas de compreensão ou de lógica.
4. **Tentativa ativa:** o plano é revisado e uma nova tentativa é realizada, agora orientada pelas correções sugeridas.

Esse ciclo é repetido até que duas respostas consecutivas se tornem semanticamente equivalentes (convergência), ou até que se atinja um número máximo de iterações. As instruções de reflexão são cuidadosamente formuladas para induzir o modelo a considerar explicitamente as dimensões de "Compreensão" e "Raciocínio", por meio de frases como:

> *"Pay attention to Comprehension and Reasoning. Check out any error and revise the plan. Then let's carry out the revised plan to solve the problem step by step."*

Contribuições e Resultados

SHP demonstrou desempenho superior em *benchmarks* de raciocínio como GSM8K, AQuA e SVAMP, superando inclusive métodos *few-shot* em determinados cenários. Em estudos de ablação, observou-se que a inclusão de instruções explícitas de reflexão aumentou significativamente a precisão dos modelos, evidenciando a eficácia da simulação da autorreflexão no processo de *prompting*.

Essa técnica representa um avanço importante em direção à engenharia de prompt baseada em modelos cognitivos e oferece caminhos promissores para tarefas que exigem robustez semântica, explicabilidade e correção iterativa.

Aplicação do SHP na Classificação de Incidentes de Segurança

Inspirada no ciclo reflexivo de Kolb, a técnica SHP foi adaptada neste trabalho para a classificação iterativa de incidentes de segurança reportados por CSIRTs e SOCs. A técnica simula um processo de autoavaliação da LLM, no qual planos intermediários são gerados, analisados e ajustados com base em observações objetivas e reflexões conscientes.

O processo inicia com a elaboração de um plano inicial para resolver o problema (classificar o incidente), seguido da execução desse plano e obtenção de uma resposta. Essa resposta é então usada como base para uma observação objetiva, sendo reavaliada por meio de um novo prompt que orienta o modelo a identificar possíveis falhas de compreensão (*comprehension issues*) ou de raciocínio (*reasoning issues*), conforme proposto por [Chen et al. 2024].

Se houver inconsistência entre respostas consecutivas, o modelo é induzido a revisar o plano anterior, gerar uma nova tentativa de classificação e repetir o ciclo. A iteração

continua até que duas respostas consecutivas sejam semanticamente equivalentes (convergência), ou até atingir o número máximo de interações permitido.

A aplicação em incidentes de segurança demonstrou ser eficaz especialmente em modelos com capacidade moderada de raciocínio abstrato, como LLaMA 4 e Grok 3. Nesses casos, o SHP permitiu a correção de erros grosseiros e a estabilização da resposta final, com acurácia média de até 86%, mesmo em cenários com alto grau de ambiguidade narrativa. Isso sugere que o ciclo de reflexão promovido pela técnica SHP é útil para lidar com dados não estruturados e contextos operacionais complexos.

Pseudocódigo SHP

Algorithm 3: Self-Hint Prompting (SHP)

Input: Incidente $INCIDENTE$, número máximo de iterações MAX_ITER, limiar de similaridade $LIMITE$
Output: Categoria classificada

$RESULTADOS \leftarrow \emptyset$;
$PLANO \leftarrow$ "Entender o problema, elaborar um plano e resolver passo a passo";
$PROMPT \leftarrow$ concatena($INCIDENTE$, $PLANO$);
$PLANO\_INTERMEDIARIO \leftarrow$ enviarParaLLM($PROMPT$);
$RESPOSTA \leftarrow$ enviarParaLLM($PROMPT$ + OUTPUT);
$CATEGORIA\_ANTERIOR \leftarrow$ extrairCategoria($RESPOSTA$);
for $i \leftarrow 0$ **to** $MAX\_ITER - 1$ **do**
    $PROMPT\_REFLEXAO \leftarrow$ concatena($INCIDENTE$, $PLANO\_INTERMEDIARIO$, $CATEGORIA\_ANTERIOR$, OUTPUT);
    $RESPOSTA \leftarrow$ enviarParaLLM($PROMPT\_REFLEXAO$);
    $CATEGORIA\_ATUAL \leftarrow$ extrairCategoria($RESPOSTA$);
    if $i = MAX\_ITER - 1$ ***ou*** *similaridade*($CATEGORIA\_ANTERIOR$, $CATEGORIA\_ATUAL$) $\geq LIMITE$ **then**
        return $RESULTADOS$;
    end
    else
        $CATEGORIA\_ANTERIOR \leftarrow CATEGORIA\_ATUAL$;
        $PROMPT \leftarrow$ concatena($INCIDENTE$, $PLANO$, $CATEGORIA\_ANTERIOR$);
        $PLANO\_INTERMEDIARIO \leftarrow$ enviarParaLLM($PROMPT$);
    end
end
return $RESULTADOS$

5. Conclusão

As técnicas *Progressive-Hint Prompting* (PHP), *Self-Hint Prompting* (SHP) e *Hypothesis Testing Prompting* (HTP) oferecem abordagens conceitualmente distintas e complementares para potencializar o desempenho de Modelos de Linguagem em tarefas de categorização de incidentes de segurança cibernética. Neste trabalho, a seleção das técnicas foi fundamentada em análise teórica, considerando critérios como independência metodológica, alinhamento com o problema proposto e diversidade cognitiva dos mecanismos de cada abordagem.

Cada técnica apresenta características únicas: o PHP estrutura o raciocínio como refinamento progressivo por meio de dicas iterativas; o SHP promove ciclos de autorreflexão com base no modelo de aprendizagem experiencial de Kolb; e o HTP adota uma lógica de verificação contrastiva baseada na formulação e testagem de hipóteses. Essa variedade de paradigmas permite explorar diferentes dimensões de inferência e compre-

ensão semântica, particularmente úteis diante da natureza ambígua e narrativa dos relatos de incidentes analisados.

A adoção dessas três técnicas,A adoção dessas três técnicas, mesmo sem a aplicação de abordagens concorrentes, garante diversidade metodológica suficiente para uma análise comparativa e contribui para o avanço do uso de LLMs na segurança cibernética. Reforça-se, assim, a importância da engenharia de *prompt* como um campo estratégico na construção de soluções automatizadas mais precisas, explicáveis e adaptáveis aos desafios reais enfrentados por CSIRTs e SOCs.

Ressalta-se, contudo, que este estudo não esgota a possibilidade de exploração de outras técnicas de engenharia de *prompt*. Estamos em processo contínuo de investigação e busca por novas abordagens que possam ampliar o portfólio de estratégias aplicáveis à análise de incidentes, bem como avaliando a combinação com diferentes taxonomias além do NIST, com o objetivo de tornar o processo de categorização ainda mais eficaz, preciso e contextualizado aos cenários reais da cibersegurança.

Referências

Chen, J., Tian, J., and Jin, Y. (2024). Self-hint prompting improves zero-shot reasoning in large language models via reflective cycle. In *Proceedings of the 46th Annual Conference of the Cognitive Science Society*.

Li, Y., Tian, J., He, H., and Jin, Y. (2024). Hypothesis testing prompting improves deductive reasoning in large language models. *arXiv preprint arXiv:2405.06707*.

Zheng, Liu, X. et al. (2023). Progressive-hint prompting improves reasoning in large language models. Online; Acesso em 07 Março 2025.