Instituto Federal do Espírito Santo – Campus Serra
Av. dos Sabiás – 330 – Morada de Laranjeiras – Serra – ES – 29166-630

Pós-graduação *stricto sensu* Mestrado Profissional em Computação Aplicada

Processo Seletivo de Alunos 2025-2

Caderno de Questões

Candidato(a): _____ **CPF:** _____

1 Instruções

Leia atentamente as instruções a seguir.

1) Você recebeu do fiscal o seguinte material:
   (a) Este **caderno de questões**, com o enunciado das 30 (trinta) questões objetivas de múltipla escolha.
   (b) O **cartão-resposta** destinado às respostas das questões objetivas formuladas nas provas.
2) Apenas o cartão-resposta será utilizado para correção. Nada que seja escrito no caderno de questões será utilizado ou considerado na correção da prova.
3) Após a conferência, o candidato deverá preencher o nome e o CPF, no espaço próprio do cartão-resposta, a caneta esferográfica na cor azul ou preta.
4) Para cada uma das questões objetivas são apresentadas 5 alternativas identificadas com as letras (a), (b), (c), (d), (e); só uma responde adequadamente à questão proposta. Você só deve assinalar **uma resposta**. A marcação em mais de uma alternativa anula a questão, mesmo que uma das respostas esteja correta.
5) Será eliminado do Processo Seletivo o candidato que:
   (a) Abandonar o local de prova sem autorização;
   (b) Ausentar-se do local de prova sem o acompanhamento do fiscal;
   (c) Mantiver conduta incompatível com a condição de candidato ou for descortês com qualquer pessoa incumbida da realização do processo seletivo;
   (d) For surpreendido, durante a realização da prova, em comunicação com outro candidato;
   (e) Sair do local da prova sem entregar o cartão-resposta;
   (f) Não permitir sua identificação;
   (g) Não atender às determinações do edital 119/2025 e de seus atos complementares;

(h) Utilizar, durante a realização das provas: chapéu, boné, livros, revistas, folhetos, impressos, anotações, calculadora ou quaisquer outros equipamentos eletrônicos de comunicação ou de consulta;

(i) Portar qualquer tipo de arma;

(j) Não assinar a lista de presença.

6) O candidato só poderá se ausentar do recinto da prova após 1 (uma) hora contada a partir do início da mesma.

7) Por motivos de segurança, o candidato só poderá levar o **caderno de questões**, depois de 2 (duas) horas contadas a partir de efetivo início da prova.

8) Recomenda-se que o candidato reserve os 20 (vinte) minutos finais para marcar seu cartão-resposta.

9) Quando terminar, entregue ao fiscal o cartão-resposta e assine a lista identificando que o cartão foi entregue.

10) O tempo disponível para esta prova de questões objetivas é de 3h00min (três horas), incluído o tempo para a marcação do seu cartão-resposta.

2 Questões de Inglês

Para responder as questões da prova de Inglês, considere o texto a seguir.

LLMs factor in unrelated information when recommending medical treatments

A large language model (LLM) deployed to make treatment recommendations can be tripped up by nonclinical information in patient messages, like typos, extra white space, missing gender markers, or the use of uncertain, dramatic, and informal language, according to a study by MIT researchers.

They found that making stylistic or grammatical changes to messages increases the likelihood an LLM will recommend that a patient self-manage their reported health condition rather than come in for an appointment, even when that patient should seek medical care.

Their analysis also revealed that these nonclinical variations in text, which mimic how people really communicate, are more likely to change a model's treatment recommendations for female patients, resulting in a higher percentage of women who were erroneously advised not to seek medical care, according to human doctors.

This work "is strong evidence that models must be audited before use in health care — which is a setting where they are already in use," says Marzyeh Ghassemi, an associate professor in the MIT Department of Electrical Engineering and Computer Science (EECS), a member of the Institute of Medical Engineering Sciences and the Laboratory for Information and Decision Systems, and senior author of the study.

These findings indicate that LLMs take nonclinical information into account for clinical decision-making in previously unknown ways. It brings to light the need for more rigorous studies of LLMs before they are deployed for high-stakes applications like making treatment recommendations, the researchers say.

"These models are often trained and tested on medical exam questions but then used in tasks that are pretty far from that, like evaluating the severity of a clinical case. There is still so much about LLMs that we don't know," adds Abinitha Gourabathina, an EECS graduate student and lead author of the study.

They are joined on the paper, which will be presented at the ACM Conference on Fairness, Accountability, and Transparency, by graduate student Eileen Pan and postdoc Walter Gerych.

Mixed messages

Large language models like OpenAI's GPT-4 are being used to draft clinical notes and triage patient messages in health care facilities around the globe, in an effort to streamline some tasks to help overburdened clinicians.

A growing body of work has explored the clinical reasoning capabilities of LLMs, especially from a fairness point of view, but few studies have evaluated how nonclinical information affects a model's judgment.

Interested in how gender impacts LLM reasoning, Gourabathina ran experiments where she swapped the gender cues in patient notes. She was surprised that formatting errors in the prompts, like extra white space, caused meaningful changes in the LLM responses.

To explore this problem, the researchers designed a study in which they altered the model's input data by swapping or removing gender markers, adding colorful or uncertain language, or

inserting extra space and typos into patient messages.

Each perturbation was designed to mimic text that might be written by someone in a vulnerable patient population, based on psychosocial research into how people communicate with clinicians.

For instance, extra spaces and typos simulate the writing of patients with limited English proficiency or those with less technological aptitude, and the addition of uncertain language represents patients with health anxiety.

"The medical datasets these models are trained on are usually cleaned and structured, and not a very realistic reflection of the patient population. We wanted to see how these very realistic changes in text could impact downstream use cases," Gourabathina says.

They used an LLM to create perturbed copies of thousands of patient notes while ensuring the text changes were minimal and preserved all clinical data, such as medication and previous diagnosis. Then they evaluated four LLMs, including the large, commercial model GPT-4 and a smaller LLM built specifically for medical settings.

They prompted each LLM with three questions based on the patient note: Should the patient manage at home, should the patient come in for a clinic visit, and should a medical resource be allocated to the patient, like a lab test.

The researchers compared the LLM recommendations to real clinical responses.

Inconsistent recommendations

They saw inconsistencies in treatment recommendations and significant disagreement among the LLMs when they were fed perturbed data. Across the board, the LLMs exhibited a 7 to 9 percent increase in self-management suggestions for all nine types of altered patient messages.

This means LLMs were more likely to recommend that patients not seek medical care when messages contained typos or gender-neutral pronouns, for instance. The use of colorful language, like slang or dramatic expressions, had the biggest impact.

They also found that models made about 7 percent more errors for female patients and were more likely to recommend that female patients self-manage at home, even when the researchers removed all gender cues from the clinical context.

Many of the worst results, like patients told to self-manage when they have a serious medical condition, likely wouldn't be captured by tests that focus on the models' overall clinical accuracy.

"In research, we tend to look at aggregated statistics, but there are a lot of things that are lost in translation. We need to look at the direction in which these errors are occurring — not recommending visitation when you should is much more harmful than doing the opposite," Gourabathina says.

The inconsistencies caused by nonclinical language become even more pronounced in conversational settings where an LLM interacts with a patient, which is a common use case for patient-facing chatbots.

But in follow-up work, the researchers found that these same changes in patient messages don't affect the accuracy of human clinicians.

"In our follow up work under review, we further find that large language models are fragile to changes that human clinicians are not," Ghassemi says. "This is perhaps unsurprising — LLMs were not designed to prioritize patient medical care. LLMs are flexible and performant enough on average that we might think this is a good use case. But we don't want to optimize a health care system that only works well for patients in specific groups."

The researchers want to expand on this work by designing natural language perturbations that capture other vulnerable populations and better mimic real messages. They also want to explore how LLMs infer gender from clinical text.

Fonte: https://news.mit.edu/2025/llms-factor-unrelated-information-when-recommending-medical-treatments-0623

1. According to the study, what effect does nonclinical information in patient messages have on LLMs?
   - (a) It improves the accuracy of treatment recommendations.
   - (b) It has no impact on the model's decisions.
   - (c) It makes LLMs more cautious in recommending self-care.
   - (d) It reduces the accuracy of the model's treatment recommendations.
   - (e) It only affects models not trained on medical data.
2. What was the impact of stylistic or grammatical changes on LLM treatment recommendations?
   - (a) They caused models to always recommend in-person visits.
   - (b) They led to more accurate diagnoses.
   - (c) They increased the likelihood of recommending self-management.
   - (d) They had no observable effect.
   - (e) They decreased the model's understanding of clinical data.
3. What is the main recommendation from the researchers regarding the use of LLMs in healthcare?
   - (a) Models should only be used for administrative tasks.
   - (b) LLMs require rigorous auditing before deployment in high-stakes settings.
   - (c) Training on medical exam questions is sufficient.
   - (d) Gender information should always be removed.
   - (e) Nonclinical language should be ignored by the model automatically.
4. Why did the researchers create perturbed copies of patient notes?
   - (a) To simulate realistic communication styles of vulnerable patient populations.
   - (b) To delete sensitive patient information.
   - (c) To test only the GPT-4 model.
   - (d) To simplify medical datasets.
   - (e) To replace clinical data with hypothetical scenarios.

5. Based on the study, which group of patients was disproportionately affected by the LLMs' inaccurate treatment recommendations?
   - (a) Patients with high technological aptitude.
   - (b) Patients with limited English proficiency.
   - (c) Male patients.
   - (d) Patients who used clean and structured language.
   - (e) Female patients.

6. According to the lead author, Abinitha Gourabathina, what is a primary limitation of the datasets typically used to train medical LLMs?
   - (a) They are too diverse and contain too much nonclinical information.
   - (b) They are not extensive enough to support medical decision-making.
   - (c) They are often "cleaned and structured," which does not reflect how people actually communicate.
   - (d) They are specifically designed to evaluate the severity of clinical cases.
   - (e) They include too many errors and inaccuracies.

7. What was the unexpected finding that led the researchers to explore how nonclinical information affects LLM judgment?
   - (a) The model was not able to draft clinical notes.
   - (b) The model performed poorly on standard medical exam questions.
   - (c) Swapping gender cues in patient notes had no effect on the model.
   - (d) Formatting errors, such as extra white space, caused meaningful changes in LLM responses.
   - (e) The LLM was more accurate when dealing with dramatic language.

8. The researchers used four different LLMs in their study. Which of the following models was specifically mentioned as being included in their evaluation?
   - (a) GPT-3.
   - (b) A smaller LLM built specifically for medical settings.
   - (c) OpenAI's GPT-5.
   - (d) An LLM trained on nonclinical information.
   - (e) The model used to draft clinical notes.

9. What does the study imply about fairness in LLM decision-making?
   (a) Seemingly minor text variations disproportionately affect certain groups, raising equity concerns.
   (b) Fairness is guaranteed if datasets are large enough.
   (c) Fairness is irrelevant in high-stakes applications.
   (d) Bias only occurs when clinical data is missing.
   (e) Fairness depends exclusively on model size.

10. What broader lesson about AI systems can be drawn from the study beyond the medical domain?
   (a) AI is unsuitable for all high-stakes applications.
   (b) AI systems should only be trained on exam-style datasets.
   (c) Stylistic variations are only relevant in medicine.
   (d) Model auditing is unnecessary outside of healthcare.
   (e) Non-task-related input features may influence AI outputs in ways that undermine fairness and safety.

3 Questões de Conhecimentos Específicos

11. Considere o seguinte trecho de código em Python 3:

```
x = 5
y = 3

z = (x % y) * (x // y)

print(z)
```

Qual será a saída?

(a) 15

(b) 10

(c) 5

(d) 2

(e) 1

12. O que será impresso na tela após a execução do trecho de código a seguir, escrito na linguagem de Programação Python 3?

```
def f(x):
    y = x + 5
    z = y * 2
    return z

print(f(f(3)))
```

(a) 16

(b) 21

(c) 42

(d) 8

(e) 20

13. Considere o seguinte código em Python 3:

```python
def soma_lista(lista):
    if not lista:
        return 0
    return lista[0] + soma_lista(lista[2:])

valores = [5, 1, 7, 9, 8]

print(soma_lista(valores))
```

Qual será a saída após a execução do código?

(a) 18

(b) 5

(c) 20

(d) 30

(e) 25

14. Qual será a saída do seguinte trecho de código Python?

```python
valores = [2, 5, 8, 11]

for i in range(len(valores)):
    if valores[i] % 2 == 0:
        valores[i] += 1
    else:
        valores[i] -= 1

print(valores)
```

(a) [3, 6, 9, 12]

(b) [1, 4, 7, 10]

(c) [2, 5, 8, 11]

(d) [2, 4, 8, 10]

(e) [3, 4, 9, 10]

15. Dada a função escrita em Python 3:

```python
def faz_algo(d):
    m = []
    for i in range(d):
        l = []
        for j in range(d):
            if i == j:
                l.append(1)
            else:
                l.append(0)
        m.append(l)
    return m

print(faz_algo(3))
```

Qual será a saída deste código?

(a) [[1, 0, 0], [0, 1, 0], [1, 0, 0]]

(b) [[1, 0, 0], [0, 0, 1], [0, 1, 0]]

(c) [[0, 1, 1], [1, 0, 1], [1, 1, 0]]

(d) [[1, 0, 0], [0, 1, 0], [0, 0, 1]]

(e) [[1], [0], [1]]

16. Qual das seguintes definições descreve corretamente um algoritmo de ordenação estável?

(a) Um algoritmo que mantém a ordem relativa de elementos com valores iguais na lista original.

(b) Um algoritmo que garante a ordenação em tempo de execução constante, O(1).

(c) Um algoritmo que, por sua natureza, sempre requer o uso de memória auxiliar adicional (espaço extra).

(d) Um algoritmo que identifica e remove automaticamente elementos duplicados durante o processo de ordenação.

(e) Um algoritmo que, durante a ordenação, prioriza a colocação dos elementos de maior valor nas posições iniciais.

17. Dado o vetor ordenado [2, 5, 8, 12, 16, 23, 38], qual será a sequência de comparações na busca binária para encontrar o elemento 16?

(a) $12 \rightarrow 8 \rightarrow 16$

(b) $12 \rightarrow 23 \rightarrow 16$

(c) $23 \rightarrow 12 \rightarrow 8 \rightarrow 16$

(d) $38 \rightarrow 23 \rightarrow 12 \rightarrow 16$

(e) $2 \rightarrow 5 \rightarrow 8 \rightarrow 16$

18. Qual estrutura de dados é mais adequada para armazenar e recuperar rapidamente um conjunto de chaves únicas, onde a ordem de inserção não é importante e a busca é a operação mais frequente?

    (a) Lista encadeada

    (b) Pilha

    (c) Fila

    (d) Árvore Binária de Busca

    (e) Tabela Hash

19. Dada a seguinte lista: [5, 8, 2, 1, 9, 4]. Qual será a lista após a segunda iteração do algoritmo de ordenação Selection Sort, considerando ordenação ascendente?

    (a) [1, 2, 8, 5, 9, 4]

    (b) [5, 1, 8, 2, 9, 4]

    (c) [1, 5, 2, 8, 4, 9]

    (d) [1, 2, 4, 8, 5, 9]

    (e) [1, 2, 5, 8, 9, 4]

20. Qual é uma das principais características do algoritmo de ordenação Merge Sort que o torna eficiente para grandes volumes de dados?

    (a) Ele ordena os elementos no local (in-place), sem necessidade de memória extra.

    (b) Ele utiliza comparações e trocas constantes para ordenar.

    (c) Ele divide o problema em subproblemas menores e combina as soluções de forma eficiente (estratégia "dividir para conquistar").

    (d) Ele garante a ordenação em tempo linear O(n) em todos os casos.

    (e) Ele é mais rápido que o QuickSort em todos os cenários.

21. Considere um grafo não direcionado com 10 vértices e 15 arestas. Qual a complexidade de tempo, considerando a análise assintótica, para percorrer todos os vértices e arestas utilizando o algoritmo de Busca em Largura (BFS) ou Busca em Profundidade (DFS)?

    (a) $O(V^2)$

    (b) O(E)

    (c) O(V)

    (d) O(V + E)

    (e) O(V log V)

22. Considere a implementação a seguir, escrita em Python 3:

```
class Pilha:
    def __init__(self):
        self.itens = []

    def empilhar(self, item):
        self.itens.append(item)

    def desempilhar(self):
        if not self.esta_vazia():
            return self.itens.pop()
        else:
            return None

    def esta_vazia(self):
        return len(self.itens) == 0

    def topo(self):
        if not self.esta_vazia():
            return self.itens[-1]
        else:
            return None

minha_pilha = Pilha()

minha_pilha.empilhar(10)
minha_pilha.empilhar(20)

minha_pilha.desempilhar()

minha_pilha.empilhar(5)

print(minha_pilha.desempilhar(), end=' ')
print(minha_pilha.topo())
```

O que será impresso na tela?

(a) 10 20

(b) 5 10

(c) 5 20

(d) 20 10

(e) 20 5

23. A matriz de adjacência de um grafo não ponderado $G = (V, E)$ com $n = |V|$ vértices é:

    (a) Uma matriz $n \times m$, onde $m = |E|$, indicando a presença de arestas.

    (b) Uma matriz $n \times n$, onde cada entrada indica a existência de uma aresta entre dois vértices.

    (c) Um vetor de tamanho n, armazenando o grau de cada vértice.

    (d) Um conjunto de listas, cada uma com os vizinhos de um vértice.

    (e) Uma árvore binária que armazena pares ordenados de vértices.

24. Considere a implementação parcial de uma árvore binária de busca em Python 3:

```python
class Node:
    def __init__(self, val):
        self.val = val
        self.left = None
        self.right = None

def inserir(root, valor):
    if root is None:
        return Node(valor)
    if valor < root.val:
        __________  # Linha 1
    else:
        __________  # Linha 2
    return root

raiz = None

for v in [10, 5, 15, 3, 7, 12, 18]:
    raiz = inserir(raiz, v)
```

    Quais linhas devem ser usadas para completar corretamente a função inserir?

    (a) Linha 1: root.left = Node(valor) e Linha 2: root.right = Node(valor)

    (b) Linha 1: root.left = inserir(root.left, valor) e Linha 2: root.right = inserir(root.right, valor)

    (c) Linha 1: root.left.append(valor) e Linha 2: root.right.append(valor)

    (d) Linha 1: root.left = valor e Linha 2: root.right = valor

    (e) Linha 1: root.left.add(valor) e Linha 2: root.right.add(valor)

25. Considerando a análise assintótica, assinale a alternativa correta sobre a comparação de complexidade entre Bubble Sort, Merge Sort e Quick Sort no pior caso:

    (a) Bubble Sort: O(n^2), Merge Sort: O(n log n), Quick Sort: O(n^2)

    (b) Bubble Sort: O(n log n), Merge Sort: O(n^2), Quick Sort: O(n log n)

    (c) Bubble Sort: O(n^2), Merge Sort: O(n^2), Quick Sort: O(n^2)

    (d) Bubble Sort: O(n), Merge Sort: O(n log n), Quick Sort: O(n^2)

    (e) Bubble Sort: O(n^2), Merge Sort: O(n), Quick Sort: O(n log n)

26. Considere a implementação parcial do algoritmo de ordenação Quick Sort em Python 3:

```python
def quicksort(lista):
    if len(lista) <= 1:
        return lista
    pivo = lista[0]
    subset1 = [x for x in lista[1:] if x < pivo]
    subset2 = [x for x in lista[1:] if x >= pivo]
    return __________  # Linha a completar
```

    Qual linha completa corretamente a função mantendo a ordenação correta?

    (a) subset1 + pivo + subset2

    (b) pivo + subset1 + subset2

    (c) subset2 + [pivo] + subset1

    (d) quicksort(subset1) + quicksort(subset2) + [pivo]

    (e) quicksort(subset1) + [pivo] + quicksort(subset2)

27. Dada a proposição: "Se chover, então o time não joga."

    Qual das seguintes proposições é logicamente equivalente à negação desta proposição?

    (a) Se não chover, então o time joga.

    (b) O time não joga se e somente se chover.s

    (c) Se o time joga, então chove.

    (d) Chove e o time joga.

    (e) Chove ou o time não joga.

28. Considere as seguintes proposições:

    P: Maria estuda.
    Q: João dorme.

    A proposição "Se Maria estuda, então João não dorme" pode ser representada como:

    (a) $p \vee \sim q$
    (b) $\sim p \wedge q$
    (c) $p \rightarrow \sim q$
    (d) $\sim (p \vee q)$
    (e) $q \rightarrow p$

29. A proposição $(p \vee q) \wedge \sim p$ é logicamente equivalente a:

    (a) $q \wedge \sim p$
    (b) $p \wedge q$
    (c) $p \vee q$
    (d) $\sim q$
    (e) $\sim (p \vee q)$

30. Considere as seguintes proposições:

    P: O software é instalado.
    Q: O computador reinicia.

    Qual das proposições a seguir é equivalente a "O software é instalado se, e somente se, o computador reinicia"?

    (a) Se o software é instalado, então o computador reinicia.
    (b) Se o computador reinicia, então o software é instalado.
    (c) O software não é instalado ou o computador reinicia.
    (d) O software é instalado ou o computador reinicia, mas não ambos.
    (e) O software é instalado quando o computador reinicia, e o computador reinicia quando o software é instalado.