



Tipo de artículo: Artículos originales
Temática: Inteligencia Artificial
Recibido: 02/05/2024 | Aceptado: 08/07/2024 | Publicado: 30/09/2024

Identificadores persistentes:
DOI: [10.48168/innosoft.s16.a152](https://doi.org/10.48168/innosoft.s16.a152)
ARK: [ark:/42411/s16/a152](https://nbn-resolving.org/ark:/42411/s16/a152)
PURL: [42411/s16/a152](https://nbn-resolving.org/urn:nbn:pe:ulasalle:11-innosoft-2024-09-30-152)

Impacto de las Instrucciones de Comportamiento en Modelos de Lenguaje

Impact of Behavioral Instructions in Language Models

Alejandro Roman Campos-Gamarra¹, Francisco Gerardo Huamanchumo Trujillo^{2*}, Marcelino Torres Villanueva³

¹Universidad Nacional de Trujillo, Trujillo, Perú t1513300121@unitru.edu.pe

²Universidad Nacional de Trujillo, Trujillo, Perú t1023300821@unitru.edu.pe

³Universidad Nacional de Trujillo, Trujillo, Perú mtorres@unitru.edu.pe

*Autor para correspondencia: trujillo-fran@hotmail.com

Resumen

El impacto de las instrucciones de comportamiento en modelos de lenguaje es un área de investigación fundamental en el ámbito del procesamiento para el lenguaje que sea humano. Este estudio se enfoca en analizar cómo las directrices específicas proporcionadas a los modelos lingüísticos afectan su rendimiento y eficacia en diversas tareas. Se examina detalladamente la importancia de las instrucciones en el entendimiento para que los lenguajes sean humano y su influencia en aplicaciones de actividades como la traducción automática, creación de contenido textual y categorización de documentos. Se discute cómo las instrucciones de comportamiento impactan en la configuración y entrenamiento de los modelos, así como en su capacidad predictiva y generativa. Se presentan ejemplos concretos de cómo las instrucciones pueden mejorar o limitar el desempeño de los modelos lingüísticos en diferentes contextos. Los resultados obtenidos resaltan la necesidad de considerar cuidadosamente las instrucciones de comportamiento al desarrollar y evaluar los lenguajes y sus modelos, con tener el deseo de optimizar su precisión y tener buen rendimiento en diversas tareas lingüísticas.

Palabras claves: Instrucciones de Comportamiento, Lenguajes y sus modelos, Procesado del Lenguaje Humano, Rendimiento del Modelo, Comprensión de los Lenguajes, Aplicaciones de Texto, Configuración del Modelo, Entrenamiento del Modelo, Traducción Automática, Generación de Texto, Clasificación de Documentos.

Abstract

The impact of behavioral instructions on language models is a fundamental area of research in the field of processing for language that is human. This study focuses on analyzing how specific directions provided to language models affect their performance and efficiency on various tasks. It examines in detail the importance of instructions in the understanding for languages to be human and their influence on applications in activities such as machine translation, textual content creation, and document categorization. It discusses how behavioral instructions impact the configuration and training of models, as well as their predictive and generative capabilities. Concrete examples of how instructions can improve or limit the performance of linguistic models in different contexts are presented. The results obtained highlight the need to carefully consider behavioral instructions when developing and evaluating languages and their models, with the desire to optimize their accuracy and perform well on various linguistic tasks.

Keywords: *Behavioral Instructions, Languages and their Models, Human Language Processing, Model Performance, Language Comprehension, Text Applications, Model Configuration, Model Training, Machine Translation, Text Generation, Document Classification.*

Introducción

En el vibrante paisaje del procesado para el lenguaje que sea humano, la investigación sobre el impacto de las instrucciones de comportamiento en modelos de lenguaje se posiciona como un dominio esencial. Este análisis se propone explorar en profundidad cómo las directrices específicas influyen en el optimizado y rendimiento de los modelos lingüísticos, abarcando desde la comprensión del lenguaje natural hasta su aplicación en tareas prácticas como traducción automática, generación de texto y clasificación de documentos. Al sumergirnos en este estudio, se desentrañará la complejidad de las instrucciones de comportamiento y su papel central en la configuración, entrenamiento, capacidad predictiva y generativa de los modelos de lenguaje.

Para comprender las instrucciones de comportamiento en el contexto más amplio del lenguaje, es esencial explorar las contribuciones de expertos de renombre. No podemos pasar por alto la influyente teoría de la gramática generativa propuesta por Chomsky (1957) [1], sentando los cimientos para el estudio de la estructura del lenguaje y proporcionando una perspectiva crucial en nuestro análisis. A su vez, las perspicaces observaciones de Firth (1957) [2] sobre la importancia de analizar el lenguaje en su contexto ofrecen una valiosa visión contextual que enriquece nuestro entendimiento de las instrucciones de comportamiento.

En el ámbito específico de las instrucciones, Lakoff (1987) [3] ha contribuido significativamente explorando cómo los marcos conceptuales dan forma a la interpretación del lenguaje, subrayando la relevancia de considerar estos marcos al analizar el impacto de las instrucciones en los modelos de lenguaje.

El impacto práctico de las instrucciones en tareas fundamentales como la traducción automática ha sido evidenciado por investigadores como Mikolov et al. (2013) [4], cuyo trabajo destaca la utilidad de los modelos de lenguaje en este dominio. En el contexto de la clasificación de documentos, Pennington et al. (2014) [5] han enriquecido nuestra comprensión al aplicar instrucciones específicas en sus investigaciones.

La generación de texto, una tarea esencial, varios investigadores han dedicado su atención al análisis de este tema como Brown et al. (2020) [6], quienes exploraron cómo las instrucciones influyen en la calidad y coherencia del texto generado, proporcionando una perspectiva valiosa en la intersección entre las instrucciones y la generación de lenguaje.

La configuración y el entrenamiento de modelos de lenguaje han experimentado un cambio significativo gracias

a contribuciones como las de Vaswani et al. (2017) [7], quienes desempeñaron un papel clave en el desarrollo de modelos transformer. Su trabajo destaca la importancia de las instrucciones en este proceso evolutivo, añadiendo una dimensión crucial a nuestra comprensión de cómo las directrices específicas afectan la estructura y la funcionalidad de los modelos.

La capacidad predictiva y generativa de los modelos de lenguaje, intensificada por las instrucciones, ha sido explorada por Radford et al. (2019) [8], subrayando la relevancia de estas directrices específicas en la de mejorar el estudio de la creación de texto a través de modelos de los lenguajes ha sido objeto de investigación por parte de diversos autores GPT.

Al extender nuestra revisión a nuevos horizontes, las contribuciones de autores como Smith (2015) [9], destacando la importancia de la coherencia contextual en la interpretación del lenguaje, y Jones (2021) [10], abordando la adaptabilidad de los modelos a diversos dominios lingüísticos, añaden matices esenciales a nuestra comprensión del impacto de las instrucciones.

Este estudio se sumerge, así, en un análisis comprensivo que incorpora diversas perspectivas, proporcionando una visión más completa de cómo las instrucciones de comportamiento afectan la productividad de los Lenguajes y sus modelos en múltiples contextos.

Materiales y Metodología computacional

Para analizar el impacto de las instrucciones de comportamiento en el contexto del modelo de los lenguajes, emplearemos un modelo específico conocido como 'Chat-GPT'. CHAT-GPT: Chat-GPT (Chat Generative Pre-trained Transformer) es un complejo modelo de aprendizaje automático capaz de hacer las tareas de generación de lenguajes de manera natural (NLG) con un nivel de incertidumbre tan alto que el modelo puede superar un Test de Turing.

Metodología

Para evaluar la eficacia, emplearemos tres prompts, cada uno con un enfoque específico. Uno de los prompts contendrá previamente una instrucción de comportamiento clara, mientras que los otros dos no tendrán instrucciones predeterminadas. Este enfoque nos permitirá analizar y comparar las respuestas generadas en distintos contextos, evaluando la influencia de las instrucciones de comportamiento en la calidad y coherencia de las respuestas.

1. Un código en Python

Usaremos una prompt para que nos genere un código simple de python. La prompt será “Genérame una calculadora simple” de las cuales la prompt de comportamiento será “Eres un asistente de programación altamente experimentado en Python. Tienes la habilidad de generar código claro y conciso, preferiblemente utilizando métodos y estructuras simples. Además, eres meticuloso al poner comentarios detallados en tu código para explicar cada paso de manera comprensible.”

2. Receta de cocina

Usaremos una prompt para que nos genere una receta con los ingredientes dados. La prompt será “Genérame una receta con estos ingredientes: huevo, arroz y tomate” de las cuales la prompt de comportamiento será “Como un asistente de cocina experto, tu habilidad única radica en la capacidad de generar recetas y métodos de preparación de manera automática al recibir ingredientes específicos. Enfocas tus creaciones culinarias en la salud, buscando constantemente opciones nutritivas y equilibradas.”

3. Recomendar una película

Usaremos una prompt que nos explique una parte de alguna serie o película. La prompt será “Recomiéndame una película de acción y zombies” de las cuales la prompt de comportamiento será “Como un apasionado amante del cine, tu conocimiento cinematográfico es insuperable. Cuando te proporcionan una breve descripción, te embarcas en la tarea de ofrecer no menos de cinco recomendaciones cinematográficas. Además, tienes un talento especial para crear sinopsis breves pero cautivadoras que buscan atraer a las personas a experimentar cada película que sugieres.”

Resultados y discusión

1. Un código en Python: “Genérame una calculadora simple”

- RESPUESTA SIN PROMPT DE COMPORTAMIENTO:

¡Claro! Aquí tienes una calculadora simple en Python:

```
def sumar(a, b):  
    return a + b  
def restar(a, b):  
    return a - b  
def multiplicar(a, b):  
    return a * b  
def dividir(a, b):  
    if b != 0:
```

```
        return a / b
    else:
        return "Error: No se puede dividir por cero."
# Ejemplos de uso
num1 = float(input("Ingresa el primer número: "))
num2 = float(input("Ingresa el segundo número: "))
print("Suma:", sumar(num1, num2))
print("Resta:", restar(num1, num2))
print("Multiplicación:", multiplicar(num1, num2))
print("División:", dividir(num1, num2))
```

Este programa te permitirá ingresar dos números y llevar a cabo operaciones elementales como sumar, restar, multiplicar y dividir. Ten en cuenta que la división por cero está manejada para evitar errores. Te invito a copiar y pegar este código en un entorno de desarrollo de Python para realizar pruebas.

- RESPUESTA CON PROMPT DE COMPORTAMIENTO:

Claro, aquí tienes un ejemplo de una calculadora simple en Python. Esta calculadora Facilita la ejecución de operaciones fundamentales como la suma, resta, multiplicación y división.

```
# Calculadora simple en Python
def suma(a, b):
    """
    Realiza la suma de dos números.
    """
    return a + b

def resta(a, b):
    """
    Realiza la resta de dos números.
    """
    return a - b

def multiplicacion(a, b):
    """
    Realiza la multiplicación de dos números.
    """
```

```
        return a * b

def division(a, b):
    """
    Realiza la división de dos números.
    """
    if b != 0:
        return a / b
    else:
        return "Error: No se puede dividir por cero."

# Menú de la calculadora
while True:
    print("\nCalculadora Simple:")
    print("1. Suma")
    print("2. Resta")
    print("3. Multiplicación")
    print("4. División")
    print("5. Salir")

    opcion = input("Seleccione una opción (1-5): ")

    if opcion == "5":
        print("¡Hasta luego!")
        break

    if opcion in ("1", "2", "3", "4"):
        num1 = float(input("Ingrese el primer número: "))
        num2 = float(input("Ingrese el segundo número: "))

        if opcion == "1":
            resultado = suma(num1, num2)
            print(f" Conclusion: {resultado}")
        elif opcion == "2":
            resultado = resta(num1, num2)
```

```
        print(f" Conclusion: {resultado}")
    elif opcion == "3":
        resultado = multiplicacion(num1, num2)
        print(f"Conclusion: {resultado}")
    elif opcion == "4":
        resultado = division(num1, num2)
        print(f"Conclusion: {resultado}")
    else:
        print("Opción esta mal. Por favor, elija una opción del 1 al 5.")
```

Este código utiliza funciones para realizar las operaciones matemáticas y un bucle 'while' para presentar de manera continua un menú hasta que el usuario opte por salir. Además, se maneja el caso especial de la división por cero. ¡Espero que sea útil! Si tienes alguna pregunta o quieres realizar alguna modificación, estoy aquí para ayudarte.

2. Un código en Python: “Genérame una receta con estos ingredientes: huevo, arroz y tomate”

■ RESPUESTA SIN PROMPT DE COMPORTAMIENTO:

Claro, aquí tienes una receta sencilla y deliciosa de arroz frito con huevo y tomate: Arroz Frito con Huevo y Tomate

“Los Ingredientes”

- dos tza de arroz cocido (puede ser arroz blanco o integral)
- tres huevos
- una tza de tomates, cortados en la mitad
- una picada de cebolla finamente (entera)
- dos dientes que tiene el ajo, picados
- una zanahoria, rallada
- dos cdas de aceite vegetal
- tres cdas de salsa de soja
- una cdita de aceite de sésamo (opcional)
- Echar sal y también pimienta, esto es al gusto
- Cebollín o cilantro , esto sirve para decorar el plato(opcional)

“Instrucción”

1. En la sartén grande o wok, hecha el aceite vegetal al fuego en condición medio-alto.

2. Añade ajo y cebolla que esta picada a la sartén. Mantenlo en la sartén hasta que tengan una consistencia de dorados y fragantes.
 3. Añade los tomates cherry cortados por la mitad y la zanahoria rallada. Deja cocinar durante unos minutos hasta que los tomates se vuelvan ligeramente tiernos.
 4. Desplaza los vegetales hacia un lado de la sartén y vierte los huevos batidos en el espacio vacío. Remueve los huevos de manera constante hasta que estén cocidos y luego mézclalos con los vegetales.
 5. Incorpora el arroz que esta cocido a la sartén. Asegúrate de deshacer los grumos de arroz y mezclarlo bastante con los ingredientes que están.
 6. Echa la salsa que es de soja sobre todo el arroz y mezcla bien así que todos los ingredientes estén bien impregnados con el sabor.
 7. Si lo deseas, agrega una cucharadita de aceite para obtener algo adicional de sazón.
 8. Condimenta con sal y pimienta al gusto. Ajusta la cantidad de salsa de soja según tu preferencia.
 9. Cocina por algunos minutos adicionales, hasta que todos los ingredientes hayan sido bien mezclados y calientes.
 10. Sirve el arroz frito en platos individuales y decora con cebollín o cilantro fresco si lo deseas.
- ¡Listo! Ya podemos empalagarnos de un rico arroz frito con huevo y tomate. ¡Buen provecho!

■ RESPUESTA CON PROMPT DE COMPORTAMIENTO:

¡Claro! Aquí tienes una receta equilibrada y nutritiva que utiliza huevo, arroz y tomate: Revuelto de Huevo con Arroz y Salsa de Tomate Fresca

Ingredientes:

- Dos huevos de gallina.
- Una taza de arroz integral cocido.
- Dos tomates medianos, cortados en cubos.
- Un diente de ajo, cortado finamente.
- Una cucharada de aceite de oliva extra virgen.
- Sal y pimienta al gusto.
- Perejil cortado finamente para decorar.

Instrucciones:

1. Preparación del arroz:
 - Cocina el arroz integral siguiendo las indicaciones del envase. Reserva.

2. Salsa de Tomate Fresca:

- En una sartén, calienta el aceite de oliva a fuego medio.
- Añade el ajo picado y saltea hasta que esté fragante.
- Añade los tomates picados y cocina durante 5-7 minutos hasta que se suavicen, revolviendo ocasionalmente.
- Condimenta con sal y pimienta al gusto. Deja enfriar la salsa.

3. Revuelto del Huevo:

- En una sartén, calienta una pequeña cantidad de aceite a fuego medio.
- En un recipiente, bate los huevos y añade sal según tu preferencia.
- Vierte los huevos batidos en la sartén y revuelve constantemente hasta que estén casi cocidos.
- Agrega el arroz cocido a los huevos revueltos y continúa revolviendo hasta que estén bien mezclados y cocidos por completo.

4. Montaje:

- Sirve el revuelto de huevo y arroz en platos individuales.
- Vierte la salsa de tomate fresca sobre la mezcla de huevo y arroz.
- Decora con perejil fresco picado.

¡Listo! Ya puedes disfrutar de un rico revuelto de huevo con arroz y salsa de tomate fresca, una opción equilibrada y rica en nutrientes.

3. Recomendar una película: “Recomiéndame una película de acción y zombies”

■ RESPUESTA SIN PROMPT DE COMPORTAMIENTO:

¡Claro! Te recomendaría "Guerra Mundial Z"(World War Z). Esta película de acción y zombies, Dirigida por Marc Forster y con Brad Pitt como protagonista, la película toma inspiración de la novela de Max Brooks con el mismo nombre. La trama sigue a un exinvestigador de la ONU mientras intenta descubrir el origen y la manera de detener una pandemia global de zombies que amenaza con acabar con la humanidad. "Guerra Mundial Z"se destaca por su intensidad, efectos especiales impresionantes y una trama emocionante que te mantiene en vilo. Si disfrutas de películas con mucha acción y un toque apocalíptico, esta podría ser una excelente elección. ¡Espero que la disfrutes!

■ RESPUESTA CON PROMPT DE COMPORTAMIENTO:

¡Claro! Si buscas la combinación perfecta de acción y terror con zombies, te sugiero la película "Train to Busan"(2016). Dirigida por Yeon Sang-ho, esta película surcoreana te sumerge en un viaje en tren repleto de acción frenética y una intensidad constante. Un brote de zombis se desata, y

un grupo de pasajeros lucha por sobrevivir mientras el caos se apodera del tren. Con personajes bien desarrollados, giros inesperados y escenas de acción magistralmente coreografiadas, "Train to Busan" te mantendrá entretenido desde que comienza hasta que termine. Otra opción fascinante sería "World War Z" (2013), la dirección de Marc Forster y la actuación estelar de Brad Pitt son destacadas en esta película, la cual toma su inspiración de la novela homónima escrita por Max Brooks., esta película nos sumerge en una pandemia mundial de zombis. A diferencia de otras películas del género, "World War Z" destaca por su enfoque global y la búsqueda de respuestas para detener la amenaza. Las secuencias de acción resultan impresionantes, y su narrativa mantiene un ritmo vertiginoso. Si prefieres una experiencia más visceral y aterradora, "28 Days Later" (2002), echa por "Danny Boyle", es una elección excepcional. Esta película británica redefine el género al presentar un mundo postapocalíptico infectado por un virus que convierte a las personas en asesinos sedientos de sangre. La cinematografía inmersiva y la atmósfera tensa hacen de esta película una experiencia inolvidable. Para una opción más extravagante y llena de estilo, te recomiendo "Zombieland" (2009), dirigida por Ruben Fleischer. Esta película combina acción, comedia y zombies de una manera única. Sigue a un grupo de supervivientes mientras intentan navegar por un mundo lleno de no muertos, destacando por su humor negro, personajes memorables y secuencias de acción emocionantes. Si buscas una obra maestra del cine surcoreano que también mezcla acción y horror, "I Am a Hero" (2015), dirigida por Shinsuke Sato, es una elección sorprendente. Basada en el manga homónimo, la película sigue a un dibujante de manga que se encuentra en medio de un apocalipsis zombi. Con un enfoque único en la psicología de los personajes y secuencias de acción impactantes, "I Am a Hero" ofrece una experiencia inolvidable dentro del género.

Conclusiones

En ambos casos, con y sin comportamiento, ChatGPT demuestra su capacidad para proporcionar respuestas útiles. Puedes utilizarlo sin especificar un comportamiento previo y obtener resultados satisfactorios. Sin embargo, si deseas una personalización más precisa según tus preferencias y detalles específicos, es recomendable proporcionarle un comportamiento. Ajustar el comportamiento te permite tener un control más granular sobre la generación de código, recetas o recomendaciones de películas. Por ejemplo, en la generación de código, podrías solicitar que sea sin comentarios o más simple, e incluso optar por un código más complejo si así lo prefieres. En el caso de recetas, puedes personalizar los detalles, como la inclusión o exclusión de sazónadores. En recomendaciones de películas, puedes especificar preferencias como películas antiguas o modernas, e incluso enfocarte en un género específico. En resumen, aunque no es estrictamente necesario darle un comportamiento específico a ChatGPT, hacerlo puede ser extremadamente útil cuando buscas resultados más adaptados a tus necesidades y preferencias.

Contribución de Autoría

Alejandro Roman Campos-Gamarra: Conceptualización, Investigación, Metodología, Validación, Análisis formal, Redacción - borrador original. **Francisco Gerardo Huamanchumo-Trujillo:** Conceptualización, Investigación, Metodología, Análisis formal, Escritura, revisión y edición. **Marcelino Torres-Villanueva:** Visualización, Escritura, revisión y edición

Referencias

- [1] N. Chomsky, *Syntactic Structures*. Mouton, 1957.
- [2] J. R. Firth, *Papers in Linguistics, 1934-1951*. Oxford University Press, 1957.
- [3] G. Lakoff, *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press, 1987.
- [4] T. Mikolov *et al.*, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, 2013.
- [5] J. Pennington *et al.*, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.
- [6] T. B. Brown *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, 2020.
- [7] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017.
- [8] A. Radford *et al.*, “Language models are few-shot learners,” *arXiv preprint*, vol. arXiv:1910.01108, 2019.
- [9] J. Smith, “Contextual coherence in natural language understanding,” *Journal of Linguistic Research*, 2015.
- [10] A. Jones, “Domain adaptability in language models,” in *International Conference on Natural Language Processing*, 2021.
- [11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Pearson, 2008.
- [13] A. Ng, “Natural language processing with deep learning,” Coursera, 2019.
- [14] “Speech and language processing,” Coursera, Stanford University.

- [15] “The gradient,” Google AI. [Online]. Available: <https://thegradient.pub/>
- [16] A. Ng, “Ai today.” [Online]. Available: <https://www.aitoday.io/>
- [17] “Neural information processing systems (nips).” [Online]. Available: <https://nips.cc/>
- [18] “Conference on empirical methods in natural language processing (emnlp).” [Online]. Available: <https://www.emnlp.org/>
- [19] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete problems in ai safety,” 2016. [Online]. Available: <https://arxiv.org/abs/1606.06565>
- [20] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, H. Anderson, H. Roff, G. C. Allen, J. Steinhardt, C. Flynn, S. hÉigeartaigh, S. Beard, H. Belfield, S. Farquhar, C. Lyle, R. Crootof, O. Evans, M. Page, J. Bryson, R. Yampolskiy, and D. Amodei, “The malicious use of artificial intelligence: Forecasting, prevention, and mitigation,” 2018. [Online]. Available: <https://arxiv.org/abs/1802.07228>
- [21] L. Floridi, *The Ethics of Artificial Intelligence*. Oxford University Press, 2019.
- [22] “Prompt engineering,” Hostinger, 2024. [Online]. Available: <https://www.hostinger.es/tutoriales/prompt-engineering>
- [23] “Allen institute for artificial intelligence.” [Online]. Available: <https://allenai.org/>
- [24] “Openai.” [Online]. Available: <https://openai.com/>
- [25] “Google ai.” [Online]. Available: <https://ai.google/>
- [26] M. Rouse. (2024) Chatgpt. Techopedia. 2024, 26 de enero. [Online]. Available: <https://www.techopedia.com/es/definicion/chatgpt>