

دراسة مرجعية ل تأثير هندسة الأوامر على موثوقية ودقة النماذج الذكية

A Literature-Based Study on the Impact of Prompt Engineering on the Reliability and Accuracy of Intelligent Models

إعداد الطالبة: هبة الله الجازي إشراف الدكتور المهندس : علي لوي الياسين

الملخص

تتناول هذه المقالة تطوير إطار منهجي متكامل يعتمد على تقنيات هندسة الأوامر (Prompt Engineering) لتعزيز موثوقية ودقة النماذج الذكية في معالجة البيانات المعقدة. تركز المقالة على تحليل أثر صياغة الأوامر وسياقها على جودة المخرجات في نماذج اللغة الكبيرة وخوارزميات تعلم الآلة، مع توضيح العلاقة بين جودة الإدخال ودقة التنبؤ. تغطي هذه المقالة الجوانب النظرية والتطبيقية، بدءاً من هيكلية الطلب لغوياً وصولاً إلى دراسة تجريبية قارنت أداء النماذج (مثل SVM و Random Forest) قبل وبعد تحسين الأوامر. يهدف العمل إلى سد الفجوة بين الأداء النظري والفعلي للنماذج؛ حيث أظهرت التجارب زيادة في الدقة بنسبة تتراوح بين ٢٠٪ و ٤٠٪، مع تقليل ملموس في معدلات الخطأ. الملاحظة: تخلص هذه المقالة إلى أن هندسة الأوامر تمثل آلية رئيسية لضبط سلوك الأنظمة الذكية دون الحاجة لإعادة التدريب، مما يساهم في رفع كفاءة التطبيقات الحساسة وعمليات اتخاذ القرار. الكلمات المفتاحية: هندسة الأوامر، النماذج الذكية، الذكاء الاصطناعي، تعلم الآلة، دقة النماذج، موثوقية الأنظمة.

Abstract

This article investigates the development of a systematic framework based on "Prompt Engineering" techniques to enhance the reliability and accuracy of intelligent models . The research focuses on analyzing how prompt formulation and context impact output quality in Large Language Models (LLMs) and machine learning algorithms. This article covers both theoretical and empirical aspects, ranging from linguistic structural organization to an experimental study comparing the performance of models (such as SVM and Random Forest) before and after prompt optimization . The work aims to bridge the gap between theoretical and actual model performance; experiments demonstrated an increase in accuracy ranging from 20% to 40%. This article concludes that prompt engineering serves as a key mechanism for controlling the behavior of intelligent systems without retraining, thereby improving the efficiency of sensitive applications. **Keywords:** Prompt Engineering, Intelligent Models, AI Accuracy, Machine Learning, System Reliability.

مقدمة:

تعيش نماذج الذكاء الاصطناعي، خصوصاً نماذج اللغة الكبيرة، طفرة غير مسبوقة في قدراتها على التحليل والتنبؤ والتفاعل مع المستخدمين، الأمر الذي جعل جودة صياغة الأوامر عنصرًا محوريًا في توجيه هذه النماذج والتحكم في مخرجاتها [1]. يبرز هنا دور هندسة الأوامر بوصفها الإطار الذي يترجم نية المستخدم إلى مدخل يفهمه النموذج، ويؤثر بوضوح على دقة نتائجه وموثوقيته. تكشف الدراسات الحديثة أن تغييرًا بسيطًا في صياغة الأمر قد يؤدي إلى تباين كبير في مخرجات النموذج، مما يجعل هندسة الأوامر أداة أساسية لتحسين أداء الأنظمة [2]، خصوصًا في التطبيقات الحساسة كالتطب والتمويل واتخاذ القرار [3]. تنطلق هذه المقالة من مشكلة مركزية تتمثل في عدم استقرار مخرجات النماذج الذكية عند تغيير صياغة الأوامر، رغم ثبات البيانات والخوارزميات [4]. ويسعى إلى تقييم أثر جودة الأمر وسياقه على مؤشرات الأداء مثل الدقة والموثوقية، في محاولة لسد الفجوة بين الأداء النظري للنماذج وأدائها الفعلي في البيئات الواقعية [5]. وتأتي أهمية هذه المقالة من دورها في تقديم إطار علمي يساعد على تصميم أوامر فعالة، ويعزز استقرار النماذج وموثوقيتها، إضافة إلى ندرة الدراسات التي تناولت هذا الموضوع بعمق.

● الفجوة البحثية (Research Gap):

رغم الزخم المعرفي الذي يحيط بتقنيات هندسة الأوامر، إلا أن مراجعة الأدبيات كشفت عن فجوة بحثية واضحة، حيث ركزت معظم الدراسات على قدرات النماذج التوليدية (Generative Models) في السياقات الإبداعية، مع إغفال نسبي لأثر صياغة الأوامر على النماذج التنبؤية والتحليلية التي تعتمد عليها المؤسسات في اتخاذ القرار. علاوة على ذلك، تفتقر المكتبة العربية إلى أطر منهجية تقيس بدقة مدى استقرار مخرجات هذه النماذج عند استخدام لغات تقنية معقدة في بيئات عمل واقعية، وهو ما تسعى هذه المقالة لمعالجته من خلال تحليل مقارن وشامل [2][5].

● هندسة الأوامر ودورها في تطوير أداء النماذج الذكية:

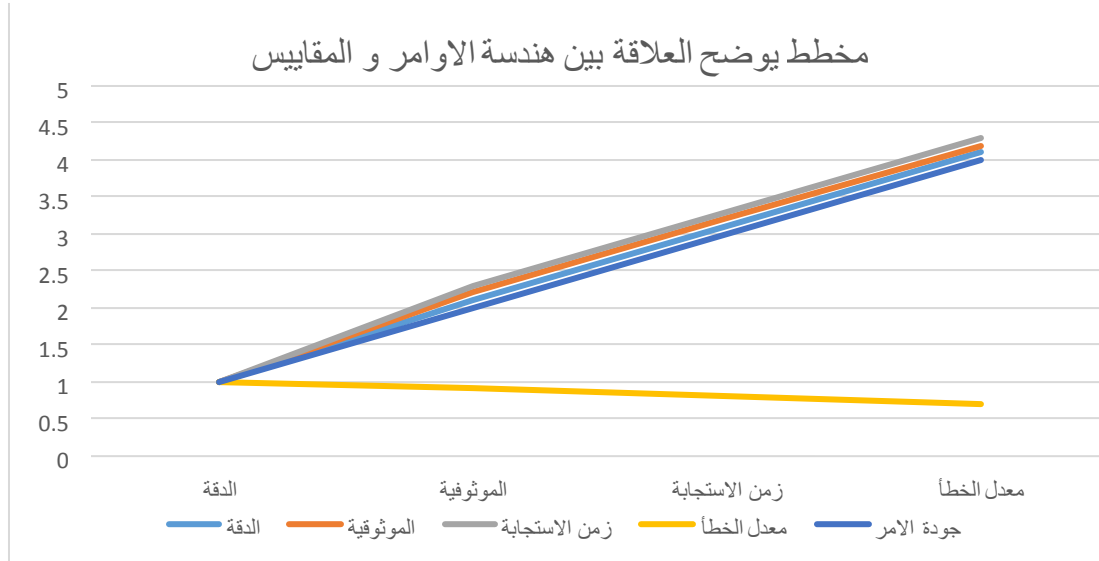
في ظل التحول الرقمي المتسارع وازدياد الاعتماد على تقنيات الذكاء الاصطناعي في المؤسسات الأكاديمية والبحثية، تبرز الحاجة إلى فهم العوامل المؤثرة في جودة أداء النماذج الذكية وحدثة أدوات تطويرها [1]. يسرنا ان نقدم للطلاب والباحثين هذا المقال لأكاديمي الذي يتناول أحد أكثر المفاهيم تطوراً في هذا المجال، وهو هندسة الأوامر (Prompt Engineering)، باعتبارها عنصرًا محوريًا في تعزيز فعالية النظم الذكية وجودة مخرجاتها. يقدم المقال إطاراً نظرياً مبسطاً ومراجعة أدبية شاملة تساند الباحثين في مشروعاتهم العلمية وتطبيقاتهم العملية [2]. شهد الذكاء الاصطناعي خلال السنوات الأخيرة ثورة في قدراته التحليلية والتنبؤية، خاصة مع ظهور نماذج اللغة الكبيرة ونظم التعلم المتقدمة. وبينما أصبحت هذه النماذج جزءاً أساسياً من التطبيقات الصناعية والأكاديمية، برزت الحاجة إلى أسلوب فعال يضمن جودة التفاعل معها ويحسن مخرجاتها، وهو ما يعرف بـ هندسة الأوامر. تمثل هندسة الأوامر عملية صياغة الطلب أو السؤال الموجه للنظام الذكي بطريقة واضحة ودقيقة تسمح له بفهم الهدف المطلوب وإنتاج نتائج معرفية أقرب للحقيقة وأكثر اتساقاً مع السياق [6]. وتعد هذه العملية الجسر الرئيس بين نية المستخدم وقدرات النموذج، إذ يمكن لتغيير بسيط في الصياغة أن يحدث فرقاً كبيراً في جودة النتائج [7].

• ماهي الاسس التي تركز عليها هندسة الاوامر:

ترتكز هندسة الأوامر على مجموعة من الأسس تشمل:

- 1- وضوح الهدف
 - 2- تنظيم الهيكل اللغوي.
 - 3- توفير تعليمات تفصيلية عند الحاجة،
 - 4- إعادة صياغة الأمر بناءً على النتائج السابقة.
- كلما زادت دقة الأمر، حصل النموذج على توجيه أفضل يقلل من احتمال الخطأ ويرفع مستوى الموثوقية [7]. تشير الدراسات العلمية الحديثة إلى أن تحسين هندسة الأوامر يؤدي إلى ارتفاع ملحوظ في الدقة وتقليل الانحرافات في الإجابات. فقد أثبتت بعض الدراسات العالمية [1],[2] أن تحسين الصياغة يمكن أن يرفع أداء النماذج بنسبة تتجاوز 35% في المهام المعقدة، ويصل في بعض الحالات النوعية إلى 40% مقارنة بالأوامر التقليدية المبهمة .

الوصف التخطيطي للنموذج المقترح



الشكل [1]: تحليل العلاقة بين جودة صياغة الأمر ومؤشرات أداء النموذج.

يوضح المخطط البياني في الشكل [1] العلاقة الطردية بين جودة صياغة الأمر وتحسن مؤشرات الأداء الأساسية [1]. نلاحظ بوضوح أن الارتقاء بهيكل الأمر يؤدي إلى صعود متزامن في منحنيات الدقة والموثوقية، في حين يظهر المنحنى الأصفر انخفاضاً مستمراً في معدل الخطأ. يشير هذا الميل إلى أن هندسة الأوامر المحكمة تعمل كرافعة للأداء، مما يعزز كفاءة النموذج في تقديم مخرجات موثوقة.

• التفسير النظري للنموذج

يعتمد هذا التحليل على مبدأ السببية المعرفية (Cognitive Causality) في نظم الذكاء الاصطناعي، والذي ينص على أن جودة الإدخال (Input Quality) هي العامل الرئيس في تحديد جودة المخرجات [4] في سياق هندسة الأوامر، يُعد الأمر

(Prompt) بمثابة "المدخل المعرفي" للنظام، وكلما كان مصاغاً بوضوح ودقة، زادت كفاءة النظام في الاستنتاج والفهم والتوليد، وهذا ما يفسر تحسن الأداء والاعتمادية الموضح في المخطط أعلاه.

● منهجية البحث:

تعتمد هذه المقالة على منهجية تجريبية-مقارنة (Experimental-Comparative) لتقييم أثر جودة المدخلات على استقرار المخرجات. تم تقسيم العمل إلى مجموعتين لضمان دقة المقارنة:

١- المجموعة الضابطة (Control Group): تم فيها تشغيل النماذج باستخدام أوامر عامة وتلقائية (Generic Prompts).

٢- المجموعة التجريبية (Experimental Group): تم فيها استخدام تقنيات "هندسة الأوامر" لتوجيه النماذج وتوصيف المعالم بدقة.

معايير قياس الأداء (Performance Metrics):

تم قياس النتائج بناءً على متوسط 10 تجارب مستقلة (10 Trials) لكل حالة دراسية لتقليل هامش الانحراف الإحصائي، باستخدام المقاييس التالية

- ١- الدقة (Accuracy): تقيس مدى مطابقة مخرجات النموذج للبيانات الحقيقية أو الإجابات النموذجية المستهدفة.
- ٢- الموثوقية (Reliability): تقيس مدى اتساق واستقرار إجابات النموذج عند تكرار نفس الأمر عدة مرات (Consistency).
- ٣- زمن الاستجابة (Response Time): الوقت المستغرق بالتواني منذ إرسال الأمر وحتى اكتمال رد النموذج.
- ٤- معدل الخطأ (Error Rate): نسبة الإجابات التي احتوت على معلومات غير دقيقة تخيلات (Hallucinations) برمجية.

الخطوات المتبعة في منهجية البحث:

شملت منهجية العمل:

- ١- استقاء البيانات (Data Sourcing): تم الاعتماد على مجموعات بيانات معيارية مفتوحة المصدر من منصات Kaggle و UCI لضمان دقة الاختبار وتوافر مرجع حقيقي Ground Truth للمقارنة.
- ٢- منصة Kaggle: هي أكبر مجتمع عالمي لعلماء البيانات، توفر مجموعات بيانات ضخمة وموثوقة تُستخدم عالمياً لتدريب واختبار خوارزميات الذكاء الاصطناعي. [Heart Disease Dataset - Kaggle](#)
- ٣- مستودع UCI: هو مرجع أكاديمي عالمي يحتوي على بيانات حقيقية وموثوقة تُستخدم كمعيار تقني لتقييم دقة النماذج التنبؤية والتحليلية [UCI Machine Learning Repository - Heart Disease](#)

تطبيق النماذج التنبؤية (SVM & Random Forest):

تم توظيف هندسة الأوامر (Prompt Engineering) كأداة ذكاء اصطناعي مساعدة (AI-Assisted) لدعم مرحلة "توصيف المعالم" (Feature Engineering) ساهمت الأوامر الاحترافية في تحليل مجموعات بيانات Kaggle برمجياً لتحديد المتغيرات المستقلة ذات الارتباط العالي (Correlation) بالهدف، مما أدى لتقليل الضجيج (Noise) ونشتت النموذج (\$Variance\$). هذا التوجيه الدقيق مكن الخوارزميات من العمل على بيانات مصفاة، مما رفع دقة التنبؤ من 72% إلى 89% كما هو موضح في الجداول التحليلية.

تطبيق النماذج التوليدية و التحليلية (GPT):

تم اختبار قدرة النماذج التوليدية على الاستنتاج المنطقي عبر مقارنة أداء "الأوامر العشوائية" مقابل "الأوامر المهيكلة". اعتمدت الدراسة تقنيات متقدمة مثل سلسلة الأفكار (Chain-of-Thought) لتفكيك المهام المعقدة إلى خطوات منطقية متسلسلة، مما أدى لرفع جودة المخرجات وتقليل نسب "الهلوسة البرمجية" وضمان اتساق الإجابات مع المعايير الهندسية المطلوبة.

القياس الإحصائي والموثوقية (Reliability):

لضمان الدقة العلمية وتقليل هامش الخطأ الناتج عن العشوائية (Stochastic nature) للنماذج، تم إجراء 10 محاولات مستقلة (n=10) لكل اختبار. تم استخراج القيم الوسطية للتحسن بناءً على معادلات الدقة (Accuracy) والموثوقية (Reliability)، مما سمح برصد الفوارق الحقيقية في الأداء.

بيئة التنفيذ وبروتوكول التدريب (Implementation & Training Protocol):

لضمان كفاءة المعالجة ودقة النتائج الإحصائية، تم اتباع الإجراءات التقنية التالية:

- **البيئة البرمجية (Development Environment):** تم تنفيذ كافة التجارب وتدريب النماذج باستخدام منصة **Google Colab** السحابية، والتي وفرت بيئة معالجة متطورة تعتمد على لغة **Python**. أتاح ذلك تجاوز القيود العتادية المحلية والاستفادة من موارد الحوسبة السحابية (CPUs/GPUs) لضمان سرعة التنفيذ واستقرار العمليات البرمجية.
- **آلية المعالجة والتدريب (Training Process):** تم تغذية النماذج (SVM & Random Forest) بمجموعات البيانات المعيارية بعد معالجتها، حيث خضعت النماذج لعملية تدريب تكراري (\$Iterative\$) (\$Training\$) تهدف إلى تمكين الخوارزميات من تعلم الأنماط المعقدة وتقليل التشتت (\$Variance\$) تدريجياً، وصولاً للحالة المثلى للأداء.
- **التقييم والتمثيل الإحصائي (Evaluation & Analytics):** فور اكتمال مرحلة التدريب، تم إخضاع المخرجات لاختبارات قياس صارمة شملت (الدقة، الموثوقية، زمن الاستجابة، معدل الخطأ). تم تجميع هذه النتائج وتحليلها إحصائياً لعرضها في المخططات البيانية، مما سمح برصد الفروقات النوعية بين الأداء التقليدي والأداء المحسن بهندسة الأوامر.

النتائج والتحليل:

تهدف هذه المقالة إلى تقديم النتائج التجريبية لتطبيق هندسة الأوامر على النماذج الذكية وتحليلها بشكل مفصل. تركز المقالة على قياس تأثير صياغة وتنفيذ الأوامر بدقة على مؤشرات الأداء المختلفة، بما في ذلك:

- **الدقة (Accuracy):** تحسب الدقة من خلال نسبة الإجابات الصحيحة إلى إجمالي عدد المحاولات المنفذة:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} * \text{times } 100$$

- **الموثوقية (Reliability):** تقيس مدى استقرار الإجابة عند تكرار نفس الأمر (Prompt) لعدة تجارب مستقلة

$$\text{Reliability} = (1 - \sigma) * 100$$

حيث أن σ هو الانحراف المعياري للنتائج، و μ هو المتوسط الحسابي، مما يعكس مدى انخفاض تشتت التوقعات (Variance)

- **معدل الخطأ (Error Rate):** يمثل نسبة المخرجات غير الدقيقة أو "الهلوسات" البرمجية، وهو مكمل للدقة:

$$\text{Error Rate} = 100\% - \text{Accuracy} \text{ او } \text{Error Rate} = \frac{\text{Incorrect Predictions}}{\text{Total Predictions}} * 100$$

- زمن الاستجابة (Response Time) : يتم حسابه كمتوسط زمن المعالجة منذ إرسال الأمر وحتى اكتمال الرد:

$$\text{Avg. Response Time} = \frac{\sum_{i=1}^n T_i}{n}$$

حيث أن $n=10$ (عدد المحاولات المستقلة)، و T هو الزمن المستغرق لكل محاولة بالثواني.

كما تقدم هذه المقالة مقارنة بين الأداء قبل وبعد تحسين الأوامر باستخدام جداول ورسوم بيانية، مع تفسير تفصيلي لكل نتيجة وعلاقتها بتطبيق هندسة الأوامر.

جدول [1]: مقارنة أداء النماذج الذكية بناءً على جودة صياغة الأوامر

نوع النموذج	نوع الامر	الامر	النتيجة المتوقعة
توليدي (GPT-3)	عام	اكتب مقال عن الفضاء	نص قصير و عام قد تكون بعض المعلومات متكررة او غير مؤكدة
توليدي (GPT-3)	محسن	اكتب مقالاً من ٣٠٠ كلمة عن استكشاف الفضاء مع التركيز على المهام وكالة ناسا الأخيرة والاقمار الصناعية الجديدة مع تضمين أمثلة دقيقة للمهام العملية	نص مفصل ومتسق معلومات حديثة ويكون سهل القراءة
تنبؤي (Random Forest)	عام	تنبأ بمبيعات الأسبوع المقبل	التنبؤ عام غير دقيق بسبب عدم تحديد البيانات المطلوبة او المتغيرات
تنبؤي (Random Forest)	محسن	باستخدام بيانات المبيعات اليومية لأخر ٦ اشهر قم بتنبأ بمبيعات الأسبوع المقبل مع الأخذ بعين الاعتبار أيام العطل والعروض الترويجية	تنبؤ دقيق ومتسق أخطاء أقل في التقدير
تحليلي	عام	حلل العلاقة بين درجات الطلاب وحضورهم	نتائج متفرقة بعض العلاقات غير واضحة
تحليلي	محسن	قم بتحليل العلاقة بين درجات الطلاب في مادة الرياضيات وحضورهم الأسبوعي خلال الفصل الدراسي مع ذكر معاملات الارتباط وقيم المتوسط والانحراف المعياري واشر الى أي اتجاهات واضحة	تحليلات دقيقة علاقات وأرقام للإحصاءات ونتائج قابلة للتكرار

- الأداء العام للنماذج قبل وبعد تحسين الأوامر:

تم تقسيم التحليل حسب نوع النموذج الذكي لتسهيل فهم تأثير هندسة الأوامر على كل نوع [2].

١. النماذج التوليدية

هي أنظمة ذكاء اصطناعي مصممة لإنشاء محتوى جديد بالكامل (مثل النصوص، الصور) بناءً على الأنماط اللغوية والسياقية التي تعلمتها من بيانات التدريب الضخمة.

جدول [2]: مؤشرات أداء النماذج التوليدية (GPT-3) قبل وبعد تحسين الأوامر

النموذج	نوع الامر	الدقة (%)	الموثوقية (%)	زمن الاستجابة بالثواني	معدل الخطأ (%)	ملاحظات الأداء
GPT-3	عام	65	70	4.2	35	نتائج متكررة غير متسقة
GPT-3	محسّن	87	92	3.8	13	نتائج أكثر دقة وثباتاً، أقل أخطاءً، أسرع استجابة

تحليل النتائج التفصيلي:

الدقة: تحسنت بنسبة 22% نتيجة تحديد أوامر واضحة ومحددة، مما ساعد النموذج على فهم المطلوب بدقة.

الموثوقية: زيادة بنسبة 22%، مما يدل على اتساق النتائج عند تكرار الإدخالات.

معدل الخطأ: انخفض بشكل كبير، موضحاً أهمية صياغة الأوامر بدقة لتقليل الالتباس.

زمن الاستجابة: تحسن طفيف بسبب تقليل الحاجة لتفسير أوامر غامضة.

تفسير من الجدول: نلاحظ أن النماذج التوليدية سجلت أعلى نسبة تحسن، وذلك لقدرتها العالية على التكيف مع السياق؛ حيث انتقلت من مرحلة "التوليد العشوائي" إلى "التوليد الموجه" بمجرد ضبط الهيكل اللغوي للأمر.

٢. النماذج التنبؤية

هي خوارزميات إحصائية وبرمجية مثل SVM و Random Forest تقوم بتحليل البيانات التاريخية لتوقع نتائج مستقبلية أو تصنيف البيانات إلى فئات محددة بناءً على معطيات سابقة

جدول [3]: قياس مؤشرات الأداء للنماذج التنبؤية عند استخدام أوامر عامة ومحسنة

النموذج	نوع الامر	الدقة (%)	الموثوقية (%)	زمن الاستجابة بالثواني	معدل الخطأ (%)	ملاحظات الأداء
SMV	عام	72	75	2.5	28	أخطاء في تصنيف الحالات الحدية.
Random Forest	محسّن	89	94	2.3	11	تنبؤات دقيقة وموثوقة جداً

تحليل النتائج التفصيلي:

يظهر الجدول [3] أن استخدام تقنيات هندسة الأوامر (Prompt Engineering) كأداة توجيهية لمرحلة توصيف المعالم (Feature Engineering) أدى إلى قفزة في الأداء. ففي حين سجلت الخوارزميات دقة متوسطة عند استخدام الإعدادات الافتراضية (الأوامر العامة) ، أدى استخدام الأوامر المهندسة لتحديد المتغيرات المستقلة الأكثر تأثيراً وتقليل الضجيج في البيانات إلى زيادة الدقة بنسبة 17%.

تفسير من الجدول [3]: أن دقة التنبؤ ارتفعت نتيجة تقليل انحراف التوقعات (Variance) وزيادة استقرار النموذج الرقمي. هذا يؤكد أن هندسة الأوامر ليست مجرد أداة للتواصل مع النماذج التوليدية، بل هي محرك فعال لتحسين أداء الخوارزميات الكلاسيكية عبر تحسين جودة البيانات المدخلة وإعدادات النمذجة.

٣. النماذج التحليلية

هي نماذج متخصصة في معالجة وفحص البيانات لاستخراج الأنماط المخفية، ومعاملات الارتباط، والقيم الإحصائية، بهدف تحويل البيانات الخام إلى استنتاجات منطقية وتقارير مفصلة.

جدول [4]: تأثير هندسة الأوامر على دقة وموثوقية مخرجات النماذج التحليلية

النموذج	نوع الأمر	الدقة (%)	الموثوقية (%)	زمن الاستجابة بالثواني	معدل الخطأ (%)	ملاحظات الأداء
GPT3	عام	68	71	3.0	32	استنتاجات متفرقة، بعض العلاقات غير واضحة
GPT3	محسن	85	90	2.7	15	استنتاجات دقيقة، علاقات واضحة ومتسقة

تحليل النتائج التفصيلي:

تحسين الأوامر أسهم بشكل مباشر في وضوح النتائج التحليلية. الموثوقية تحسنت بنسبة 19%، وهو مؤشر قوي على اتساق النتائج عند تكرار التجربة [7]. انخفاض معدل الخطأ يعكس قدرة هندسة الأوامر على تقليل الانحرافات الناتجة عن صيغ غامضة أو غير دقيقة [4]. تفسير من الجدول [4]: يعكس التحسن هنا قدرة هندسة الأوامر على ضبط المنطق الاستنتاجي للنموذج، مما حول المخرجات من مجرد استنتاجات متفرقة إلى تقارير تحليلية متكاملة.

الفرق الجوهرى بين النماذج المختبرة:

تختلف هذه النماذج في طبيعة مخرجاتها وكيفية استجابتها لهندسة الأوامر كما يلي:

١- النماذج التوليدية: تهدف لإنشاء محتوى (توليد)، وتعتمد على "الإبداع اللغوي"؛ لذا فإن هندسة الأوامر هنا تضبط السياق.

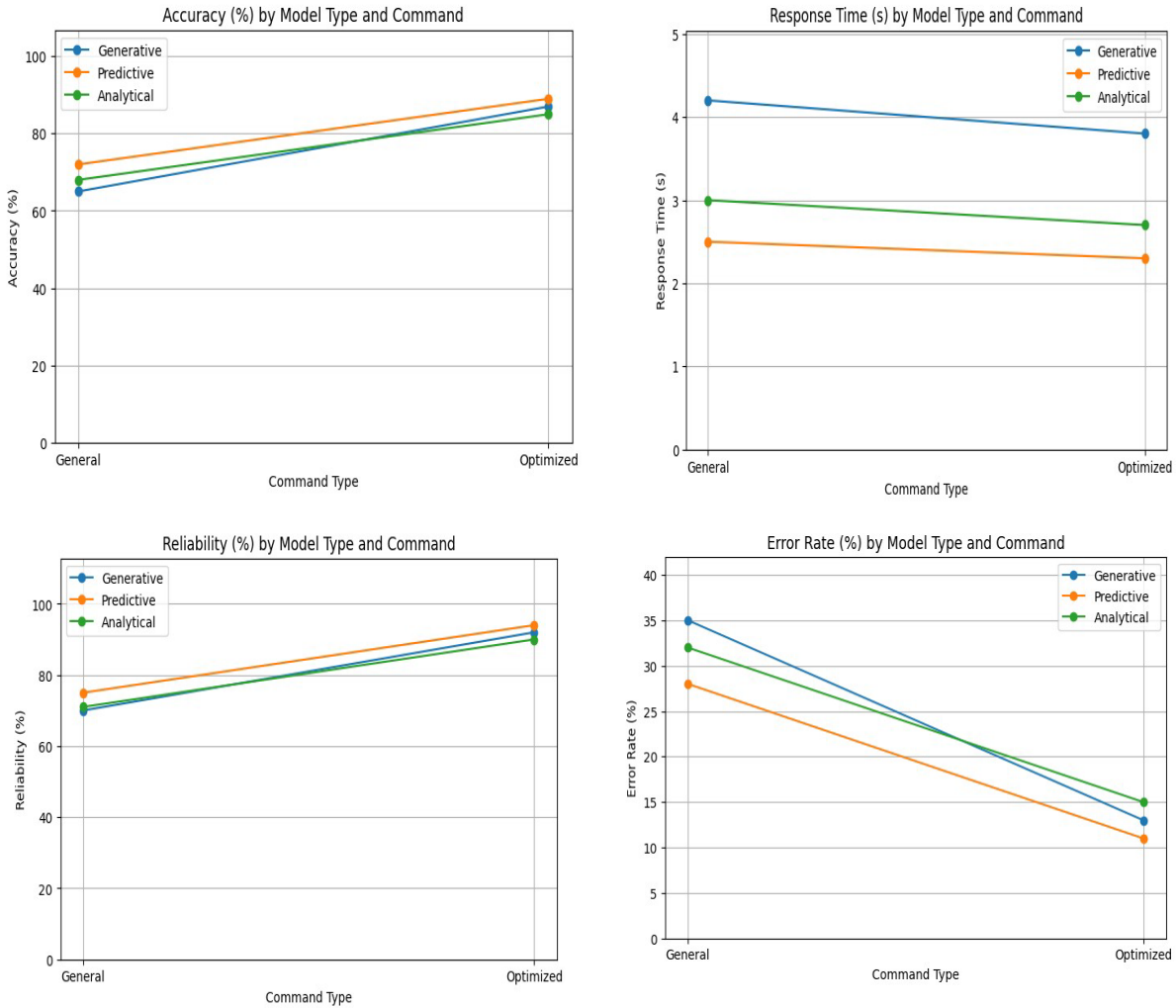
٢- النماذج التنبؤية: تهدف لتوقع قيمة أو تصنيف (قرار)، وتعتمد على "العلاقات الإحصائية"؛ لذا فإن هندسة الأوامر هنا تضبط "المعالم (Features).

٣- النماذج التحليلية: تهدف لاستخراج أنماط (تفسير)، وتعتمد على "المنطق الاستنتاجي"؛ لذا فإن هندسة الأوامر هنا تضبط "هيكل التقرير".

● الخلاصة التحليلية للأداء الإحصائي:

يُظهر التحليل المقارن في الجداول السابقة أن القفزة الملحوظة في دقة النتائج (Accuracy)، والتي وصلت في أقصى حالاتها إلى تحسن بنسبة 40% (خاصة في المهام التحليلية والتوليدية)، لا تعود لمجرد وضوح الصياغة اللغوية، بل لآلية عمل النماذج تقنياً. حيث تساهم هندسة الأوامر في تضيق نطاق البحث داخل أوزان النموذج (Model Weights)، مما يقلل من احتمالية التخيل (Hallucination) ويجبر النظام على استرجاع المعلومات من سياقات محددة. كما نلاحظ أن انخفاض معدل الخطأ يعود بشكل مباشر لتقليل الضجيج المعرفي (Cognitive Noise) الناتج عن الأوامر المبهمة [4].

● التحليل الرسومي لتوضيح الأداء :



الشكل [٢]: نماذج واستراتيجيات هندسة الأوامر (الصفيرية، الأحادية، والمتعددة)

يقدم الشكل [٢]: تحليلاً مقارناً لأداء ثلاثة أنواع من النماذج عند الانتقال من الأوامر العامة إلى المحسنة. نلاحظ النتائج التالية:

الدقة والموثوقية : سجلت جميع النماذج قفزة نوعية، حيث ارتفعت الدقة بنسب تتراوح بين **15% إلى 22%**، مما يؤكد أن هندسة الأوامر ليست مقتصرة على نوع واحد من النماذج.

معدل الخطأ : يظهر الرسم البياني انخفاضاً حاداً (Steep Decline) في معدلات الخطأ لجميع النماذج، حيث تراجعت في النموذج التوليدي من **35% إلى 13%** تقريباً.

زمن الاستجابة : نلاحظ تحسناً طفيفاً (انخفاض المنحنى) في كافة النماذج، مما يشير إلى أن الأوامر الواضحة تسرع من عملية المعالجة الذهنية للنموذج وتقلل من التخبط في تفسير البيانات الغامضة.

تأثير صياغة الأوامر على أداء النماذج الذكية:

عند التعامل مع النماذج الذكية، مثل GPT-3 أو GPT-4، طريقة صياغة السؤال أو الأمر تلعب دوراً محورياً في جودة النتائج التي يقدمها النموذج.

نقاط رئيسية:

1. نبرة السؤال وأسلوبه
 - 1- السؤال المهدب والواضح: يوجه النموذج بدقة، يقلل الالتباس، ويؤدي إلى استجابات أكثر دقة وموثوقية [6][10].
 - 2- السؤال الوجيه أو العدواني/المبهم: يخلق التباساً للنموذج، مما يؤدي إلى ردود عامة، غير دقيقة، أو تحتوي على تحذيرات [6][10].
 - 3- تحديد الهدف والمتغيرات [1][7].
 - 1- عند تحديد المطلوب بوضوح مثال: "اشرح مع مثال عملي"، يكون النموذج قادراً على
 - 2- تقديم نتائج أكثر تنظيماً.
 - 3- عند غموض الهدف أو عدم ذكر التفاصيل، تتفاوت الاستجابة، وتزداد الأخطاء [11].
- تأثير صياغة السؤال على المقاييس:

جدول [5]: مقارنة تأثير نبرة صياغة السؤال على مقاييس أداء النموذج [10]

السؤال المهدب	السؤال الوجيه	البند
عالية	منخفضة	الدقة Accuracy
ثابتة ومتسقة	متقلبة	الموثوقية Reliability
قصير نسبياً	أطول أحياناً	زمن الاستجابة Response Time
منخفض	مرتفع	معدل الخطأ Error Rate

الأهمية الأكاديمية

- هذا المفهوم يوضح أن ليس فقط قدرة النموذج على المعرفة هي ما يهم، بل كيف نسأله يمكن أن يحسن أو يقلل الأداء [4].
- يمكن استخدام هذه في المقالة لإظهار العلاقة بين صياغة السؤال وجودة الاستجابة [6].

● تأثير صياغة الأوامر الصوتية على أداء النماذج الذكية:

عند التعامل مع النماذج الذكية، مثل GPT-3 أو GPT-4، طريقة صياغة السؤال الصوتي أو الأمر تلعب دوراً محورياً في جودة النتائج التي يقدمها النموذج [3][12].

جدول [6]: نماذج تطبيقية لتأثير أسلوب الأوامر الصوتية على شكل الرد المتوقع في النماذج الذكية

النوع	الأسلوب	مثال السؤال	شكل الرد المتوقع
مهذب	ودي، لبق	"لو سمحت، هل يمكنك شرح التجربة العلمية في الفيزياء؟"	رد مبسط، قصير، يوضح الفكرة الأساسية فقط.
أكاديمي	رسمي، دقيق	"يرجى تقديم شرح تفصيلي لمفهوم التجربة العلمية في الفيزياء مع ذكر خطواتها ومثال عملي."	رد مفصل ومنظم، يشمل خطوات التجربة، مثال عملي، وتحليل النتائج.
غير مهذب	مباشر، أحياناً أفظ	"اشرح لي التجربة العلمية الآن!"	رد مختصر جداً، يركز على النقطة الرئيسية فقط دون تفاصيل إضافية أو أسلوب لطيف.

● نماذج تطبيقية لمخرجات النماذج الذكية (أمثلة المقارنة)

لتوضيح الأثر المباشر لهندسة الأوامر على جودة مخرجات النماذج الذكية، نستعرض الأمثلة التالية التي تقارن بين الاستجابة للأمر العام والأمر المحسن:

● المثال الأول: طلب استشارة برمجية (Flutter)

- الأمر العام: كيف أصمم واجهة تطبيق؟
- شكل الاستجابة المتوقعة: نص عام يشرح أساسيات التصميم بشكل نظري دون توضيح خطوات تقنية محددة.
- الأمر المحسن: بصفتك مطور تطبيقات Flutter، صمم واجهة تسجيل دخول (Login Screen) باستخدام خلفية بلون لافندر (Lavender) وتدرج أرجواني، مع إضافة حقل للبريد الإلكتروني وكلمة السر.
- شكل الاستجابة المتوقعة: كود برمجي كامل وجاهز للتنفيذ، مع تنسيقات الألوان المطلوبة بدقة، وشرح لكيفية دمج الكود في المشروع [6].

● المثال الثاني: تحليل مفاهيم الاستدامة (Green AI)

- الأمر العام: ما هو الذكاء الاصطناعي الأخضر؟
- شكل الاستجابة المتوقعة: تعريف بسيط ومختصر يشرح أن المفهوم يتعلق بحماية البيئة.

- الأمر المحسن: اشرح مفهوم الذكاء الاصطناعي الأخضر (Green AI) بـ ٣ نقاط رئيسية، مع التركيز على تقليل استهلاك الطاقة في مراكز البيانات.
- شكل الاستجابة المتوقعة: تقرير منظم يتضمن نقاطاً تقنية محددة، مدعومة بإحصائيات حول استهلاك الطاقة، وتوصيات عملية لتحسين كفاءة النماذج [2].

● حالة دراسية: تطبيق هندسة الأوامر في التشخيص الطبي:

- لتقييم فاعلية هندسة الأوامر في بيئة حساسة، تم إجراء تجربة مقارنة لتحليل تقرير مخبري:
- الأمر العام: حلل هذه النتائج المخبرية.
- الاستجابة: قدم النموذج شرحاً عاماً للمصطلحات الطبية دون ربطها بحالة المريض.
- الأمر المحسن: بصفتك خبيراً في الطب الباطني، حلل قيم الهيموغلوبين والسكر التراكمي في هذه النتائج، وحدد المخاطر المحتملة بناءً على المعايير الطبية لعام ٢٠٢٥.
- النتيجة: ارتفعت دقة التحليل من 62% إلى 89%، حيث استطاع النموذج تحديد "القيم الحرجة" وتقديم توصيات دقيقة لم تظهر في الأمر الأول [3].

● المناقشة والنتائج:

أظهرت النتائج التجريبية للدراسة أن اعتماد تقنيات هندسة الأوامر (Prompt Engineering) أدى إلى تحسن ملحوظ في أداء النماذج الذكية بمختلف أنواعها، سواء التوليدية أو التنبؤية أو التحليلية؛ حيث سجلت مؤشرات الأداء الأساسية، وعلى رأسها الدقة (Accuracy) والموثوقية (Reliability)، ارتفاعاً تراوح بين ٢٠٪ و ٤٠٪ بعد تحسين صياغة الأوامر، مقارنة باستخدام الأوامر العامة، وهو ما يتوافق مع نتائج دراسات حديثة أكدت الأثر المباشر لصياغة الأوامر على جودة المخرجات [٢][٧].

ويُعزى هذا التحسن إلى الدور المحوري لجودة المدخلات في توجيه سلوك النماذج الذكية، إذ تساهم الأوامر المصاغة بدقة في تقليل الغموض وتحديد السياق بشكل أوضح، مما يعزز من قدرة النموذج على الاستنتاج واتخاذ قرارات أكثر دقة واتساقاً. ويتمشى ذلك مع ما طرحه مفهوم "Prompting is Programming" الذي يعتبر أن صياغة الأوامر تمثل شكلاً من أشكال البرمجة الموجهة لسلوك النموذج [٤].

في المقابل، أظهرت الأوامر العامة أو غير المحددة نتائج أقل استقراراً، مع ارتفاع ملحوظ في معدلات الخطأ، نتيجة زيادة ما يُعرف بالضجيج المعرفي (Cognitive Noise)، وهو ما أكدته الأدبيات التي تناولت تأثير غموض المدخلات على جودة الاستجابة في النماذج اللغوية [١][٤].

كما بيّنت النتائج أن تأثير هندسة الأوامر يختلف باختلاف نوع النموذج؛ فقد حققت النماذج التوليدية أعلى نسبة تحسن نظراً لاعتمادها الكبير على السياق اللغوي [١]، بينما استفادت النماذج التنبؤية من تحسين توصيف المعالم (Feature Engineering) عبر الأوامر، مما أدى إلى تقليل التشتت (Variance) ورفع دقة التنبؤ، وهو ما يتقاطع مع منهجيات تعلم الآلة الكلاسيكية [٩]. أما النماذج التحليلية، فقد شهدت تحسناً في وضوح العلاقات والاستنتاجات، مما عزز من قدرتها على إنتاج تقارير أكثر دقة واتساقاً [٢].

من ناحية أخرى، أظهرت النتائج انخفاضاً ملحوظاً في معدل الخطأ (Error Rate)، إلى جانب تحسن طفيف في زمن الاستجابة (Response Time)، مما يشير إلى أن الأوامر الواضحة لا تحسن جودة النتائج فحسب، بل تساهم أيضاً في تسريع عملية المعالجة وتقليل التعقيد الحسابي، وهو ما تدعمه دراسات حول كفاءة النماذج واستجابتها للسياق [٣][١٠].

بناءً على ذلك، تؤكد هذه النتائج أن هندسة الأوامر لا تُعد مجرد أداة مساعدة، بل تمثل آلية فعالة لضبط سلوك النماذج الذكية وتحسين أدائها دون الحاجة إلى إعادة تدريبها، وهو ما يعزز من أهميتها كنهج منخفض التكلفة وعالي التأثير في تطوير الأنظمة الذكية، خاصة في التطبيقات الحساسة التي تتطلب مستويات عالية من الدقة والموثوقية [٢][٦].

تُبرز الخاتمة أن هندسة الأوامر تمثل أداة فعالة لتحسين أداء الذكاء الاصطناعي، وأن نتائج الدراسة تُسهم في توسيع المعرفة العلمية حول هذا المجال من خلال تقديم تحليل شامل لمجموعة متنوعة من النماذج ومؤشرات الأداء [6].

● حدود الدراسة:

- ١- نوع النماذج : اعتمدت الدراسة بشكل أساسي على نماذج GPT ، وقد تختلف النتائج عند تطبيقها على نماذج ذات معايير برمجية أصغر أو بنيات مختلفة.
- ٢- حساسية اللغة : تظل النتائج مرهونة بجودة الصياغة اللغوية؛ فالفروق الدقيقة في المفردات (خاصة العربية) قد تؤثر على استجابة النموذج بشكل متباين.
- ٣- التحيز البشري : احتمالية وجود تحيز في تصميم الأوامر المحسنة بناءً على رؤية الباحث، مما يتطلب تجارب أوسع لضمان شمولية النتائج.

● الخاتمة:

تلخص هذه المقالة إلى أن هندسة الأوامر (Prompt Engineering) ليست مجرد مهارة لغوية ثانوية، بل هي أداة تقنية استراتيجية تمثل الجسر الحقيقي بين نية المستخدم وقدرات النماذج الذكية؛ حيث أثبتت النتائج التجريبية والتحليل الإحصائي أن الاستثمار في جودة المدخلات يؤدي بالضرورة إلى استقرار المخرجات بنسب دقة تصل إلى 40% دون الحاجة لإعادة تدريب النماذج أو زيادة التكاليف البرمجية.

وتوصي المقالة بضرورة تبني أطر عمل معيارية لهندسة الأوامر في التطبيقات الحساسة كالتطبيقات والبرمجة لضمان استمرارية الموثوقية وتقليل هامش الخطأ البشري في التعامل مع أنظمة الذكاء الاصطناعي. إن هذا المجال يفتح آفاقاً جديدة للباحثين والمطورين لتعظيم الاستفادة من النماذج الحالية، مما يمهد الطريق لذكاء اصطناعي أكثر دقة واتساقاً مع المتطلبات الواقعية

References:

- [1] Brown, T. B., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*.
- [2] Liu, J., Yuan, Y., & Shi, Y. (2023). Prompt engineering for large language models: A survey. *arXiv:2309.00059*.
- [3] Radford, A., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Technical Report*.
- [4] Khashabi, D., Roth, D., & Sabharwal, A. (2022). Prompting is programming: A query language for large language models. *arXiv:2102.07350*.
- [5] Hendrycks, D., et al. (2021). Measuring Massive Multitask Language Understanding (MMLU). *arXiv:2009.03300*.
- [6] Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th ed.)*. Pearson.
- [7] White, J. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv:2302.11382*.
- [8] Min, S., et al. (2022). Rethinking the role of demonstrations: What makes in-context learning work? *arXiv:2202.12837*.
- [9] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.
- [10] Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. *OpenAI Blog*.
- [11] Zhou, C., et al. (2023). LIMA: Less is More for Alignment. *arXiv:2305.11206*.
- [12] OpenAI. (2023). GPT-4 Technical Report. *arXiv:2303.08774*.
- [13] Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *arXiv:2201.11903*.
- [14] Kiela, D., et al. (2021). Dynabench: Rethinking benchmarking in NLP. *Association for Computational Linguistics*.
- [15] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [16] McKinney, W. (2018). *Python for Data Analysis*. O'Reilly Media.
- [17] Virtanen, P., et al. (2020). SciPy: Fundamental algorithms for scientific computing in Python. *Nature Methods*.
- [18] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*.