

نظرة عامة على الذكاء الاصطناعي القائم على الوكالة

Overview on Agentic AI

الاستاذ ربيع ع. قبلاوي

تعريف بالذكاء الاصطناعي القائم على الوكلاء (Agentic AI)

الذكاء الاصطناعي القائم على الوكلاء يشير إلى أنظمة الذكاء الاصطناعي (غالبًا مبنية على نماذج لغوية كبيرة) التي تتمتع بقدرتها على فهم نية المستخدم، والوعي بالسياق، والقدرة على تنفيذ إجراءات بناءً على اتخاذ قرارات مستقلة نيابةً عن المستخدم.

أهمية الذكاء الاصطناعي القائم على الوكلاء

- القدرة على أتمتة سير العمل المعقدة وحل المشكلات متعددة الخطوات.
- تطبيقات في العالم الحقيقي: المساعدون الشخصيين، تحليل البيانات، الروبوتات، وغيرها.

- المفاهيم الأساسية وعناصر البناء
- أنظمة الوكلاء المتعددة والتعاون
- بروتوكول MCP
- المكتبات والأدوات
- إنشاء وكيل عن طريق موقع n8n

التعريف بالوكيل و مكوناته

- الوكيل هو كيان مستقل يتمتع بالقدرة على إدراك البيئة المحيطة به، وفهم الأوامر أو الأهداف، واتخاذ قرارات وتنفيذ إجراءات بشكل ذاتي لتحقيق تلك الأهداف، دون تدخل بشري مباشر في كل خطوة. ويتكون الوكيل من:
 - النموذج اللغوي الكبير (LLM) كـ "عقل" أو محرك التفكير.
 - عقلية/شخصية الوكيل
 - الأدوات: "الأيدي والعيون" للوكيل للتفاعل مع العالم.
 - الذاكرة: ذاكرة قصيرة وطويلة الأمد للسياق.

المكون الأول: النموذج اللغوي الضخم

جوهر الوكيل هو نظام لغوي ضخم, تعمل النماذج اللغوية الضخمة على فهم طلب المستخدم من جهة, ومن جهة ثانية, يمتلك النموذج القدرة على اختيار الأداة الأنسب من بين مجموعة أدوات متاحة لديه – مثل أدوات البحث، الحساب، التفاعل مع قواعد بيانات. علاوة على ذلك، لا يعمل النموذج بشكل منعزل، بل يقوم بتنسيق عمل المكونات الأخرى في النظام. النموذج يعمل كمنسق ذكي بين الأجزاء المختلفة. المهام باختصار:

- للتواصل مع الأدوات التقنية.
- يخطط لتسلسل الخطوات اللازمة.
- يعيد تقييم الحالة بناءً على النتائج المرحلية.
- ثم يقدم الرد النهائي للمستخدم بطريقة مفهومة ومتراصة.

المكون الثاني: العقلية \ شخصية الوكيل

هذا المكون يُعدّ أحد الأجزاء الأساسية في بناء الوكيل الذكي، وهو عبارة عن ملف يحتوي على مجموعة من الإعدادات والتعليمات التي تُحدد كيفية تفاعل النموذج اللغوي مع المستخدم والبيئة المحيطة، من خلال هذا الملف، يمكن ضبط سلوك النموذج اللغوي من حيث أسلوب الردود، درجة الرسمية أو العفوية في الخطاب، ونوعية اللغة المستخدمة سواء كانت فصحي أو لهجة محلية، أو حتى لغة أجنبية. كما يُتيح هذا المكون إمكانية تمرير معلومات وسياقات سابقة إلى النموذج، مثل أهداف المستخدم أو نتائج تفاعلات سابقة، مما يمنحه قدرة أكبر على الفهم والاستجابة بطريقة أكثر ذكاءً وفعالية.

المكون الثالث: الأدوات (Tools)

تُعدّ الأدوات بمثابة “الأيدي والعيون” التي يستخدمها النموذج اللغوي لتنفيذ المهام والتفاعل مع العالم الخارجي. فبينما يُشكّل النموذج اللغوي (LLM) “العقل المفكر”، فإن الأدوات هي الوسائط التي تمكّنه من تجاوز حدود النص والتواصل المجرد، ليؤدي أدوارًا عملية أكثر تعقيدًا، تقنيا الأدوات هي واجهات أو دوال برمجية يمكن للوكيل استخدامها لإنجاز مهام محددة مثل:

- إجراء عمليات حسابية معقدة.
- البحث في الإنترنت أو قواعد البيانات.
- تنفيذ أوامر برمجية أو استدعاء واجهات برمجة التطبيقات (APIs).
- التفاعل مع مستندات، جداول بيانات، أو ملفات.
- إدارة مهام متعددة أو التفاعل مع بيانات مادية (كالروبوتات).

كيفية عمل الأدوات داخل الوكيل

عندما يتلقى النموذج اللغوي طلبًا من المستخدم، فإنه:

1. يفهم النية من خلال التحليل اللغوي للسؤال.
2. يُقرر ما إذا كان يستطيع الإجابة اعتمادًا على معرفته الداخلية، أو يحتاج إلى استخدام أداة خارجية.
3. ينفذ الأداة المناسبة عبر استدعاء الواجهة أو تنفيذ الوظيفة.
4. يعالج الناتج من الأداة ويقدمه للمستخدم ضمن السياق المناسب.

يتم دمج الأدوات عادة من خلال إطار عمل يربط النموذج اللغوي ببيئة خارجية قادرة على تنفيذ الأوامر. ومن أشهر هذه الأطر:

- **LangChain**
- **OpenAI Function Calling**
- **Autogen من Microsoft**
- **LlamaIndex**

المكون الرابع: الذاكرة (Memory)



- في بنية الوكلاء الذكيين، تُعد الذاكرة من المكونات الحيوية التي تميز الوكيل العادي عن وكيل قادر على التعلم والتكيف والاستمرارية. فكما يحتاج الإنسان إلى تذكر السياقات والمعلومات السابقة ليتمكن من التفاعل بفعالية، يحتاج الوكيل الذكي إلى آلية لحفظ المعلومات وتوظيفها بما يتناسب مع احتياجات المستخدم وسياق المهمة. هناك نوعان من الذاكرة، طويلة الأمد (**Long-Term Memory**) وقصيرة الأمد (**Short-Term Memory**)

الذاكرة قصيرة الأمد (Short Term Memory)

- تمثل المعلومات التي تُخزن خلال جلسة واحدة فقط.
- تُستخدم لمتابعة الحوار أو تنفيذ سلسلة من التعليمات ضمن نفس المحادثة.
- تُفقد عادة بعد انتهاء الجلسة أو عند إعادة تشغيل النظام.

الذاكرة طويلة الأمد (Long Term Memory)

- تحتفظ بالمعلومات لفترات ممتدة، وقد تشمل:
 - بيانات حول تفضيلات المستخدم.
 - نتائج مهام سابقة.
 - ملفات أو مصادر مرجعية.
 - ملاحظات مستخلصة من المحادثات السابقة.
- تُستخدم لتوفير تجربة أكثر تخصيصًا واستمرارية في التعامل.

نماذج وأمثلة على استخدام الذاكرة

- مساعد شخصي ذكي يتذكر جدولك، أسلوبك في الرد، والمصطلحات التي تفضلها.
- وكيل بحث وتحليل يتذكر البيانات التي قام بجمعها سابقًا لتجنب التكرار.
- نظام دردشة طبي يحتفظ بتاريخ المريض وسوابقه المرضية لمتابعة دقيقة.

الذاكرة من الناحية التقنية

- تُستخدم قواعد بيانات أو ملفات منظمة لتخزين واسترجاع البيانات.
- يتم ربط الذاكرة بالنموذج اللغوي عبر واجهات مثل:
 - LangChain Memory
 - Vector Stores (مثل Chroma و Pinecone)
 - Redis / PostgreSQL للتخزين الهيكلي أو المؤقت.
- بعض الأنظمة تعتمد على تقنيات الاسترجاع السياقي الذكي (Contextual Retrieval) لحقن المعلومات ذات الصلة في سياق النموذج عند الحاجة.

ما وظيفة الذاكرة قصيرة الأمد في الأنظمة الذكية؟

أ) حفظ ملفات المستخدم وملاحظاته على المدى الطويل

ب) متابعة الحوار الحالي وتنفيذ التعليمات ضمن نفس الجلسة

ج) أرشفة البيانات الطبية

د) تحليل مشاعر المستخدم عبر الزمن

ما وظيفة الذاكرة قصيرة الأمد في الأنظمة الذكية؟

أ) حفظ ملفات المستخدم وملاحظاته على المدى الطويل

ب) متابعة الحوار الحالي وتنفيذ التعليمات ضمن نفس الجلسة

ج) أرشفة البيانات الطبية

د) تحليل مشاعر المستخدم عبر الزمن

الإجابة: ب, متابعة الحوار الحالي وتنفيذ التعليمات ضمن نفس الجلسة

مراحل عملية التفكير لدى الوكيل (Chain of thoughts)

1. فهم الهدف (Goal Understanding)
2. تحليل المهمة (Task Analysis)
3. تحديد الموارد (Tool & Memory Selection)
4. التخطيط (Planning)
5. التنفيذ المرحلي (Step-by-Step Execution)
6. تقييم النتائج

فهم الهدف (Goal Understanding)

- يبدأ الوكيل بمحاولة تفسير طلب المستخدم.
- يستخدم النموذج اللغوي لفهم النية والسياق والنتيجة المتوقعة.
- مثال: إذا قال المستخدم “حلل موقعي الإلكتروني”، يستنتج الوكيل أن المطلوب تحليل أداء الموقع، وليس فقط قراءة محتواه.

تحليل المهمة (Task Analysis)

- تحليل المهمة هو المرحلة التي يقوم فيها الوكيل الذكي بتفكيك الطلب الأساسي أو الهدف العام إلى مجموعة من المهام الفرعية المحددة والواضحة. هذه الخطوة تُعتبر محورية في عملية التفكير، لأنها تُحوّل الغموض أو التعميم في الطلب إلى خطوات قابلة للتنفيذ عمليًا.
- مثال: "تحسين الموقع" يمكن أن يتحول إلى:
 - a. تحليل الأداء التقني (السرعة – الحماية).
 - b. تحسين محركات البحث (SEO).
 - c. تقييم التصميم وتجربة المستخدم.

تحديد الموارد (Tool & Memory Selection)

بعد أن يُحلّل الوكيل الذكي الهدف ويفكّكه إلى مهام فرعية واضحة، تأتي مرحلة **تحديد الموارد** التي سيعتمد عليها لتنفيذ هذه المهام. وتشمل هذه الموارد:

- **الأدوات (Tools):** مثل الآلات الحاسبة، أدوات البحث، واجهات برمجة التطبيقات، أدوات تحليل البيانات، وغيرها.
- **الذاكرة (Memory):** المعلومات المخزنة مسبقًا سواء قصيرة أو طويلة الأمد، والتي قد تكون مفيدة في استكمال المهمة أو توفير السياق.
- أهمية هذه الخطوة: اختيار الطريقة الأكثر كفاءة لإنجاز المهمة.
- تقليل الحاجة إلى تكرار أو إعادة جلب بيانات.
- ضمان تفاعل ديناميكي وذكي يعتمد على السياق السابق وليس فقط على التعليمات الحالية.

التخطيط المنطقي والتسلسل (Planning)

يُرتَّب الخطوات بشكل منطقي ومتسلسل

بعد أن يُحدِّد الوكيل المهام الفرعية ويختار الموارد المناسبة، يبدأ بعملية التخطيط لتنفيذ هذه المهام. وهنا، لا يعتمد الوكيل على تنفيذ عشوائي أو ارتجالي، بل يُرتَّب الخطوات بحسب الأولوية، التبعية، والتسلسل المنطقي.

ما الذي يُؤخذ بعين الاعتبار في الترتيب؟

1. العلاقات بين الخطوات:
بعض المهام تعتمد على نتائج مهام سابقة. على سبيل المثال، لا يمكن للوكلاء تحليل نتائج تجربة إلا بعد تنفيذ التجربة نفسها.
2. الأهمية والأولوية:
يُبدأ عادة بالخطوات الأساسية أو التي تؤثر على بقية المراحل.
3. الموارد المتوفرة:
يُراعى توقيت الأدوات والبيانات، فمثلاً لا يُنفَّذ بحثاً قبل التأكد من توفر الإنترنت أو الوصول إلى API.

التنفيذ المرحلي (Step-by-Step Execution)

بعد أن يُخطط الوكيل الذكي مسار العمل ويحدد ترتيب المهام، يبدأ بتنفيذها بشكل متسلسل ومنهجي. لا يقوم الوكيل بمحاولة تنفيذ كل المهام دفعة واحدة، بل يتّبع نهج “خطوة بخطوة”، حيث يُنفذ كل مهمة على حدة، مع مراعاة السياق والاعتماد على نتائج المراحل السابقة.

لماذا التنفيذ المتسلسل مهم؟

- لأن بعض المهام تعتمد على مخرجات سابقة.
- يمنح الوكيل مرونة في التفاعل والتكيف بعد كل خطوة.
- يسمح بتقييم دقيق للأداء المرحلي، مما يعزز جودة النتيجة النهائية.

الهدف: "أكتب مقالاً عن الذكاء الاصطناعي".

الخطوات:

1. تحديد الموضوع والمحاور.
 2. البحث عن معلومات حديثة (باستخدام أدوات).
 3. وضع مخطط للمقال.
 4. الكتابة الأولية.
 5. التدقيق والتحسين.
- كل خطوة تعتمد على السابقة، ويجب تنفيذها بهذا الترتيب.

تقييم النتائج (Result Evaluation)

بعد أن ينفذ الوكيل مهمة أو سلسلة من المهام، تأتي مرحلة تقييم النتائج، وهي الخطوة التي يقيم فيها الوكيل مدى نجاحه في تحقيق الهدف المطلوب، ويقرر ما إذا كانت المهمة اكتملت، أو تحتاج إلى تعديل، أو إعادة تنفيذ.

ما الهدف من تقييم النتائج:

- التأكد من أن المخرجات صحيحة وذات جودة مقبولة.
- التحقق من أن الهدف الأساسي قد تحقق بالفعل.
- تحديد ما إذا كانت هناك حاجة إلى مزيد من الخطوات أو التعديلات.
- منع تقديم نتائج خاطئة أو غير مكتملة للمستخدم.

هذه تقنية هجينة تسمح للنموذج بالتفكير واتخاذ إجراء في الوقت نفسه، بطريقة تكرارية.

يبدأ النموذج بتفسير جزئي، ثم يُجري فعلًا (مثل استخدام أداة)، ثم يُفكر مجددًا بناءً على الناتج، وهكذا.

دورة ReAct:

1. التفكير (Reasoning): "لا أملك كل المعلومات."
2. الإجراء (Acting): "سأبحث في الإنترنت عن السعر الحالي للعملة."
3. التحليل (Reasoning): "الآن عرفت السعر، يمكنني الحساب."
4. النتيجة: "إليك الإجابة..."

الفائدة:

- تمكين النموذج من التفاعل مع الأدوات بطريقة ديناميكية.
- مفيد جدًا في المهام التفاعلية والمعتمدة على مصادر خارجية.

المهمة: "هل هذا الخبر صحيح؟"

- يفكر: "لا أملك مصدرًا داخليًا يؤكد صحته."
- يُنفذ: يستخدم أداة بحث للوصول إلى أخبار حديثة.
- يُفكر مجددًا: "وجدت المصدر، وهو موثوق."
- يستنتج: "نعم، الخبر صحيح بناءً على المصدر الفلاني."

هل هذه الخطوات **ReAct** أم سلسلة تفكير؟

فكرت: أحتاج إلى معرفة حالة الطقس في بيروت.

الإجراء: استعلام API لحالة الطقس في بيروت

الملاحظة: الطقس مشمس.

فكرت: إذا لا داعي لأخذ مظلة.

الجواب النهائي: الطقس مشمس، لا حاجة لمظلة.

هل هذه الخطوات **ReAct** أم سلسلة تفكير؟

فكرت: أحتاج إلى معرفة حالة الطقس في بيروت.

الإجراء: استعلام API لحالة الطقس في بيروت

الملاحظة: الطقس مشمس.

فكرت: إذا لا داعي لأخذ مظلة.

الجواب النهائي: الطقس مشمس، لا حاجة لمظلة.

الإجابة: ReAct, السبب: النموذج هنا "يفكر" → يتصرف → يلاحظ → يعيد التفكير"، وهو النمط الأساسي لتقنية ReAct، حيث يتفاعل مع البيئة باستخدام أدوات خارجية

تعاون الوكلاء المتعددين (Multi-Agent Collaboration)

تعاون الوكلاء المتعددين يشير إلى أنظمة يعمل فيها أكثر من وكيل ذكي بشكل منسق لأداء مهام معقدة أو متعددة الجوانب لا يستطيع وكيل واحد تنفيذها بكفاءة. لماذا نحتاج تعاون الوكلاء؟

- توزيع المهام: كل وكيل مختص بجزء معين (تحليل، تنفيذ، مراجعة...).
- التوازي: إنجاز عدة مهام في وقت واحد.
- التحكم بالمجالات المعرفية: وكلاء بمهارات مختلفة (محاسبة، برمجة، تسويق).
- تحسين الدقة والجودة: من خلال مراجعة متبادلة وتكامل في القرارات.

- التعاون التسلسلي (Sequential): كل وكيل يُكمل عمل السابق.
- التعاون الموازي (Parallel): يعمل الوكلاء بالتزامن ثم يدمجون النتائج.
- التعاون التفاوضي (Negotiation): وكلاء يتفاوضون لاختيار أفضل مسار.

مشروع إنشاء تطبيق جديد:

- وكيل للمسح السوقي.
- وكيل لتحليل المنافسين.
- وكيل لبناء النموذج الأولي.
- وكيل لتوليد خطة تسويقية.

الإنسان في الحلقة (Human-in-the-Loop)

“الإنسان في الحلقة” هو أسلوب تصميم يضع الإنسان كعنصر إشرافي أو تداخلي ضمن دورة عمل الوكيل الذكي، لضمان الدقة، الأمان، والقرار السليم.

أدوار الإنسان:

- التوجيه: تحديد الأهداف وتعديلها أثناء التنفيذ.
- المراجعة: مراجعة النتائج أو الموافقة على قرارات الوكيل.
- التدخل: إيقاف أو إعادة توجيه الوكيل عند الضرورة.
- التدريب والتحسين: تغذية النموذج بملاحظات لتطوير أدائه.

متى نحتاج الإنسان في الحلقة

- في الحالات الحساسة (طب – قانون – قرارات مالية).
- عندما يكون مستوى الثقة بالوكيل غير كافٍ.
- في المهام التي تتطلب حكمًا أخلاقيًا أو سياقًا عميقًا.

دمج الإنسان في الحلقة لا يُقلل من قيمة الوكيل، بل يُعزز كفاءته ويضمن توجيهه في الاتجاه الصحيح ضمن منظومة ذكية مسؤولة.

مكتبات وأدوات خاصة بتصميم الوكلاء

- LangChain
- OpenAI Function Calling
- Microsoft Autogen
- LlamaIndex
- OpenAI Agents_SDK
- Servants
- موقع n8n

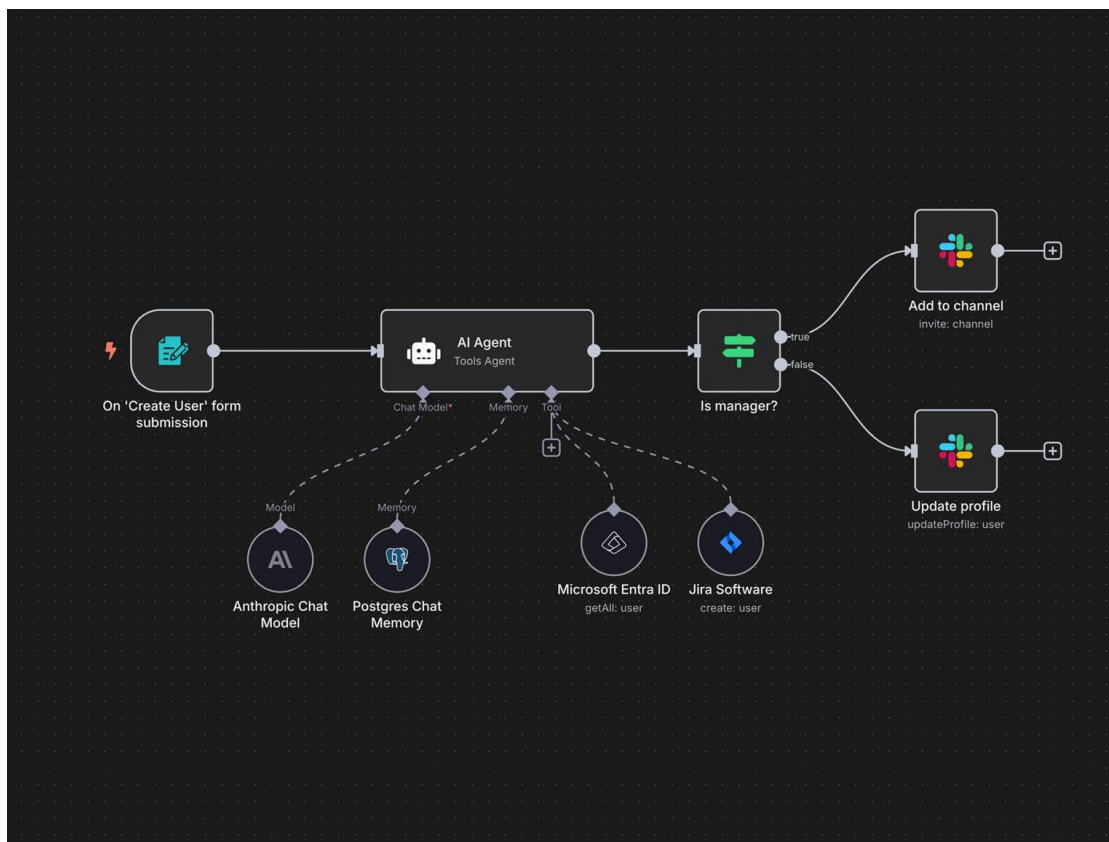
Langchain, Langgraph و Langflow

الأداة	الوصف	الاستخدام الرئيسي
LangChain	وأدوات مختلفة LLMs إطار عمل لتصميم سلاسل منطقية من العمليات باستخدام	بناء وكلاء، أدوات تفكير، استرجاع السياق
LangFlow	باستخدام السحب والإفلات Chains لتصميم الـ LangChain واجهة رسومية مبنية على	تسريع بناء النماذج والتجريب البصري
LangGraph	(LLMs) لإدارة سير عمل مرّن ومعدّد باستخدام Graph نظام يعتمد على الرسم البياني (	(Loops تصميم عمليات ذات تشعبات وتكرارات)

بروتوكول MCP – التحكم في سلوك الوكلاء

هو بروتوكول يهدف إلى تنظيم وتنسيق سلوك الوكلاء المتعددين في نظام واحد، لضمان التفاعل المنهجي وتحقيق الأهداف بكفاءة. وظائف البروتوكول الأساسية:

- تحديد الأدوار والمهام لكل وكيل.
- تنظيم سير الحوار والقرارات بين الوكلاء.
- منع التضارب والتكرار في تنفيذ المهام.
- توزيع الموارد والأدوات بطريقة ذكية.



n8n هو منصة أتمتة مفتوحة المصدر تتيح ربط التطبيقات والخدمات بسهولة لإنشاء سير عمل آلي بدون الحاجة لبرمجة معقدة.

شكرا لكم