١. الهدف من المشروع

يهدف المشروع إلى تصميم المكونات الأساسية لوكيل محادثة افتراضي ChatBot يراعي خصوصية المعلومات في بنيته، ويتيح للمستخدمين إجراء المحادثات والحصول على أجوبة مرتكزة على مصادر معلومات محددة مسبقاً.

٢. المتطلبات الوظيفية وغير الوظيفية

فيما يلي المتطلبات الوظيفية وغير الوظيفية للمشروع:

١.٢. المتطلبات الوظيفية

يجب على النظام:

١. أن يستخرج محتوى المعطيات النصية غير المهيكلة unstructured data مثال pdf, word, text.
٢. أن يتيح فهرسة المحتوى النصي للمعطيات غير المهيكلة.
٣. أن يتيح إمكانية الاسترجاع من المعطيات غير المهيكلة.
٤. أن يتيح إمكانية الاسترجاع من المعطيات المهيكلة مثال قواعد المعطيات المهيكلة.
٥. أن يتيح إمكانية الاسترجاع من المعطيات شبه المهيكلة semi-structured.
٦. أن يعالج استعلام المستخدم ويقدّم الجواب المدعّم بالوثائق اللازمة.
٧. أن يراعي سياق المحادثة أي الاستعلامات السابقة ضمن عملية تحضير الجواب.

٢.٢. المتطلبات غير الوظيفية

تشمل المتطلبات الغير وظيفية للمشروع ما يلي:

١. واجهة مستخدم سهلة الاستخدام.
٢. زمن إجابة النظام أقل من ٢٥ ثانية.
٣. إدارة وصول المستخدمين إلى حساباتهم.

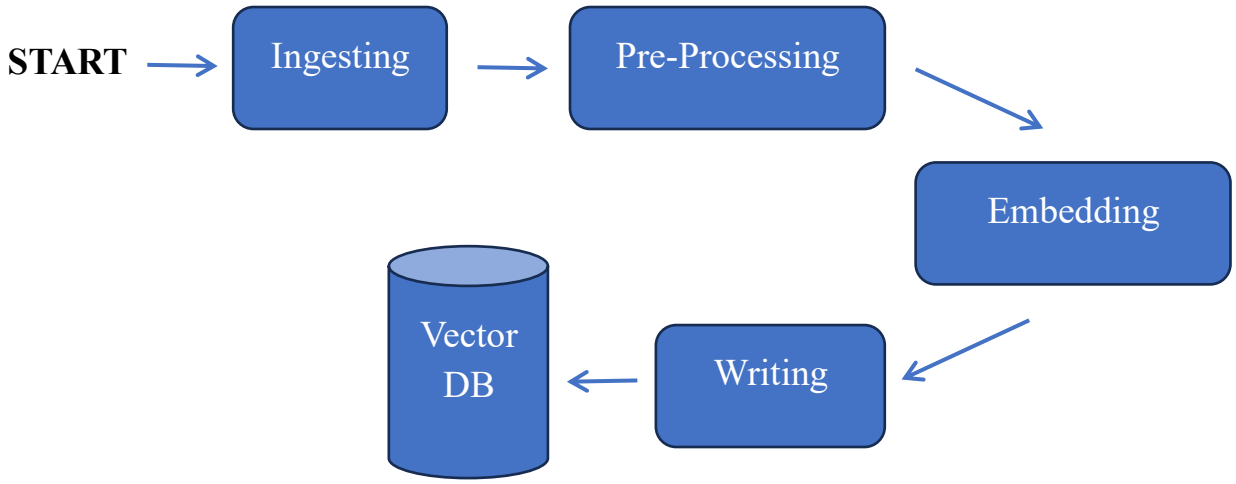
٤. إدارة وصول المستخدمين إلى محادثاتهم.

٣. تصميم النظام

يعمل النظام بمعمارية نظم RAG (Retrieval Augmented Generation) التي تعمل بمرحلتين، مرحلة الفهرسة ومرحلة الإجابة، فيما يلي مخطط عام لكلا المرحلتين يتبع كل منهما تفاصيل المكونات الجزئية ضمن سير عملية كل مرحلة، ومخطط لنظام التقييم المستخدم وتصميمه.

١.٣. مرحلة الفهرسة

تمر مرحلة الفهرسة بالعمليات التالية:



١. تحميل الملف Ingesting: إنشاء وثيقة تحمل المحتوى النصي للملف.

٢. المعالجة الأولية Pre-Processing: تشمل تنظيف محتوى الوثيقة وتجزئتها إلى قطع بطول مناسب.

٣. التضمين Embedding: ويشمل تضمين محتوى كل وثيقة للحصول على متجه يعبر عن المعنى الدلالي لها.

٤. الحفظ Writing: تخزين الوثيقة في قاعدة معطيات متجهية Vector Database.

١.١.٣ عملية تحميل الملف Ingesting

تقوم هذه العملية من خلال مكون Ingestor باستخراج محتوى الملف النصي مع مراعاة نوع هذا الملف، والنظام يدعم الاستخراج من الأنواع التالية (pdf, word, text)، لا تتضمن هذه المرحلة تجزيئاً للمحتوى النصي ويعامل النص معاملة كتلة نصية واحدة من أجل عدم إدخال عوامل غير مرغوب بها (مثل عدد الصفحات) في عملية تجزيء النص لاحقاً.

٢.١.٣ عملية المعالجة الأولية للملف Pre-Processing

تتضمن هذه العملية عمل مكونين بشكل أساسي وهما المنظّف Cleaner والمجزّئ Splitter، يقوم المنظف بالتخلص من الفراغات والسطور ضمن النص والتي تنجم عن عملية الاستخراج، وعند انتهائه يقوم المجزّئ بتقطيع النص إلى قطع chunks تراعي طولاً محدداً، تحتاج طريقة التجزيء وحجم القطعة الواحدة إلى تعيين فبالنسبة لطريقة التجزيء فهناك طريقتين أساسيتين، التقسيم حسب نمط فاصل مثل الفراغ من أجول الحصول على تجزيء على مستوى الكلمات أو النقطة وحرف نهاية السطر للحصول على تجزيء على مستوى الأسطر ، والطريقة الثانية هي بحسب الجُمَل حيث يقوم نموذج تعلم عميق متخصص بتجزيء النص إلى جمل محافظاً على المعنى الدلالي لكل جملة. أما حجم القطعة فيحدد بمراعاة أمرين، سعة سياق نموذج التضمين المستخدم والمعنى الدلالي للقطعة.

٣.١.٣ عملية التضمين Embedding

تستخدم هذه العملية نموذج تضمين محدد مسبقاً لتحويل القطع النصية لمتجهات دلالية تخزن ي نفس الغرض، ومسألة اختيار نموذج التضمين تحتاج إلى مراعاة الاختلافات بين نماذج التضمين، ليس من حيث البنية وطريقة العمل فحسب بل بالمهمة التي تم تدريب نموذج التضمين من خلالها، فتوجد نماذج تضمين مدربة على العربية فقط وأخرى متعددة اللغات، وتوجد نماذج تضمين دربت بمهمة مشابهة النصوص وأخرى لمهمة استرجاع المعطيات.

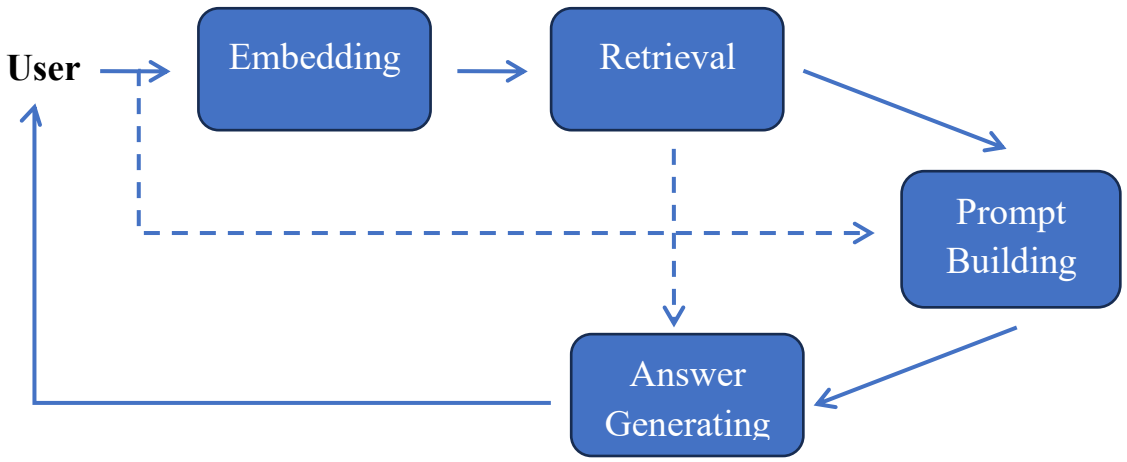
٤.١.٣ عملية الكتابة Writing

يقوم في هذه العملية مكون الكاتب Writer بتخزين الوثائق وتضميناتها ضمن قاعدة معطيات متجهيّة تراعي تضمين هذه الوثائق عند عملية الاسترجاع منها، يمتلك الكاتب أيضاً سياسة تكرار تساعد في التعامل مع الوثائق الموجودة

مسبقاً وحفظ الوثائق الجديدة ولها ثلاث قيم: تجاهُل (عدم حفظ الوثيقة) Skip تبديل (استبدال الوثيقة الجديدة بالقديمة) Replace والفشل (إيقاف عملية الكتابة وإطلاق خطأ) Fail.

٢.٣. مرحلة الإجابة

تمر مرحلة الإجابة بالخطوات التالية:



١. التضمين Embedding: تشمل تضمين استعلام المستخدم.

٢. الاسترجاع Retrieval: تشمل استرجاع الوثائق النصية المتعلقة باستعلام المستخدم.

٣. بناء الأمر Prompt Building: تستخدم هذه المرحلة قالب أمر معد مسبقاً Prompt Template وتقوم بدمج الوثائق المسترجعة واستعلام المستخدم فيه من أجل تمريرها لاحقاً للمولد.

٤. توليد الجواب Answer Generating: توجيه الأمر الذي تم بناءه إلى نموذج لغوي لتوليد جواب مناسب بالاعتماد على الوثائق المسترجعة.

١.٢.٣ عملية التضمين Embedding

تقوم هذه العملية بتضمين الاستعلام بنفس نموذج التضمين المستخدم في مرحلة الفهرسة، وهذا ضروري للحصول على متجه يعبر عن ذات التعابير الدلالية لمتجهات الوثائق.

٢.٢.٣ عملية الاسترجاع Retrieval:

خلال هذه العملية يحصل استرجاع الوثائق حيث تُستخدم خوارزمية Approximate Nearest Neighbor للوصول لأفضل K من التضمينات المشابهة لاستعلام المستخدم ومن ثم استرجاع الوثائق المقابلة لهذه التضمينات، يتم تحديد قيمة K قبل تنفيذ عملية الاسترجاع.

٣.٢.٣ عملية بناء الأمر Prompt Building:

تستخدم هذه العملية كل من استعلام المستخدم والوثائق المسترجعة ضمن قالب أمر Prompt Template أجل بناء أمر Prompt يساعد على توليد جواب للمستخدم، تم تصميم قالب الأمر بالاعتماد على نمط RACE في بناء الأوامر للنماذج اللغوية الذي يحتوي أربع معايير يجب على الأمر أن يحققها للحصول على جودة جواب أفضل من النموذج اللغوي، وتشمل المعايير:

- الدور Role: تحديد الدور الذي سيلعبه النموذج اللغوي في الإجابة.
- الفعل Action: وتشمل المهمة الذي سيقوم بها النموذج وتعليمات القيام بها.
- السياق Context: وتشمل الفرضيات والمتغيرات التي سيقوم النموذج بأخذها بالاعتبار أثناء توليد الجواب.
- التنفيذ Execute: وتشمل ملاحظات نهائية على شكل الإجابة أو قيود يجب الالتزام بها، كما يمكن ذكر أمثلة ضمن هذا القسم لإرشاد النموذج خلال التوليد.

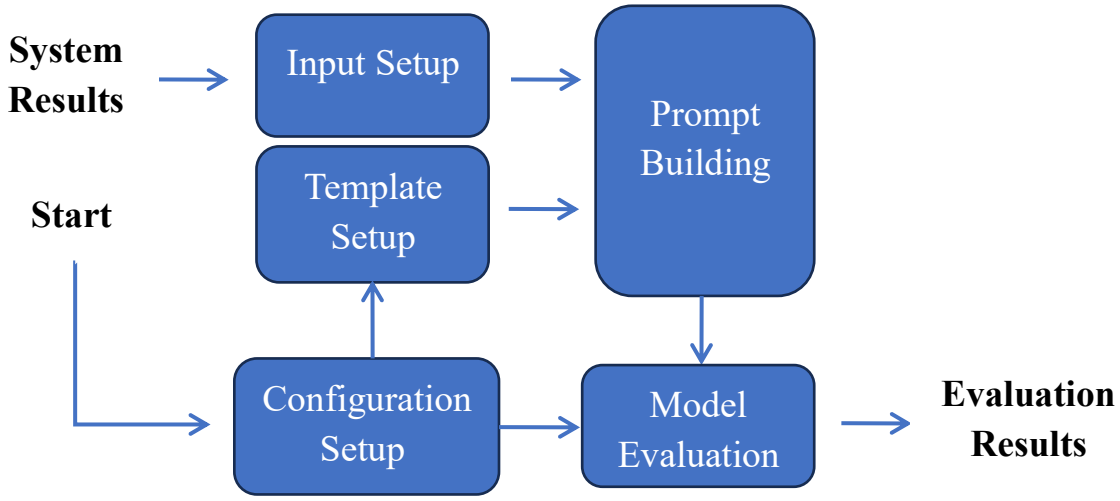
بناء الأمر بشكل صحيح يزيد اتساق النموذج اللغوي في تقديم إجاباته ويساعده على توليد أجوبة أدق وذات جودة أعلى.

٤.٢.٣ عملية توليد الجواب Answer Generating:

تستخدم هذه العملية نموذج لغوي Language Model للإجابة، يستلم الأمر القادم من المرحلة السابقة ويقوم بتوليد جواب بناءً عليه.

٣.٣. مخطط نظام التقييم

يعتمد النظام في تقييمه على ستة مقاييس لتقييم للأداء النهائي، الأول مدمج مع النظام في حين يتبع الخمسة الأخرى نظاماً ذو منهجية مختلفة هي منهجية "النموذج كحكم" أو LLM-as-Judge حيث يقوم نموذج لغوي كبير أو صغير مُختص بتقييم النظام، يعرض المخطط التالي العمليات التي يقوم بها النظام لأداء مهمته.



١. تحميل إعدادات التقييم Configuration Setup: خطوة مهمة لتحديد المقاييس التي يجب تنفيذها والنموذج الذي سيقوم بالتقييم.

٢. إعداد القوالب Template Setup: بعد أن تم تحديد المقاييس والنماذج المطلوبة، يتم إحضار قوالب الأوامر الخاصة بهذه المقاييس والمناسبة للنموذج المستخدم.

٣. إعداد المعطيات Input Setup: تقوم بتنظيم النتائج بطريقة مناسبة كمدخلات للمرحلة القادمة.

٤. بناء الأوامر Prompt Building: تحضير أوامر التقييم باستخدام قوالب المقاييس وقيم نتائج النظام.

٥. تقييم النموذج Model Evaluation: تنفيذ التقييم.

١.٣.٣ مقاييس التقييم المستخدمة

يستخدم النظام مقياساً مدمجاً لسهولة حسابه مقارنة بالمقاييس الأخرى وهو مقياس تشابه الإجابة الدلالي (SAS) أو Semantic Answer Similarity الذي يستخدم نموذج تضمين لحساب تشابه جيب التمام cosine

similarity بين تضمين كل من الإجابة المولدة من النظام والإجابة المرجعية المخزنة. أما بالنسبة للمقاييس الخمسة البقية فهي:

١. مقياس أهمية السياق Context Relevance: يقيس مدى دقة استرجاع المعلومات مع الاستعلام. ويتوفر منه مقياسين مختلفين Context Recall وContext Precision.
٢. مقياس الأمانة Faithfulness: يقيس ما إذا كانت الإجابة المولَّدة تعكس بدقة المعلومات الواردة في المستندات المسترجعة، وتحدد الاتساق بين المحتوى المولَّد والوثائق المسترجعة لمنع الهلوسة.
٣. مقياس صلة الاستجابة Response Relevancy: يقيس مدى صلة الإجابة المولَّدة بالاستعلام الأصلي.
٤. الصحة المرجعية Factual Correctness: يقيس مدة دقة صحة الإجابة مقارنة بجواب مرجعي موثوق.

٢.٣.٣ نموذج التقييم

تعتمد منهجية تقييم "النموذج كحكم" LLM-as-Judge على نموذج لغوي محكّم يقيّم نتائج النظام. وقد يكون نموذجاً لغوياً كبيراً ذو أداء عالي مثل OpenAI GPT 4.1 وClaude Sonnet 4، أو نماذج صغيرة مدربة من أجل مهمة تقييم النتائج، ففعاليتها مستندة على مبدأ أن "تقييم صحة الإجابة أسهل من توليدها" التي تم إثباتها من خلال تقييمات لهذه النماذج على مجموعات معطيات مرجعية أبرزتها كمنافسة للنماذج الكبيرة السابقة.

٤. التنفيذ والاختبارات

يتناول هذا الفصل التنجيز البرمجي للنظام، الأدوات المستخدمة وأسلوب العمل ثم الاختبارات والنتائج.

١.٤ الأدوات المستخدمة في تنجيز النظام

جرى تنجيز النظام بلغة بايثون Python لكونها اللغة الأكثر استخداماً في نظم المبنية على التعلم العميق، واستعملنا مجموعة من المكاتب أهمها:

١. Google Colaboratory (Colab): بيئة سحابية بلغة بايثون ساعدت على تطوير واختبار النظم بفضل العتاد الذي توفره.

٢. Haystack-ai إطار عمل مفتوح المصدر لبناء مكونات النظم المعتمدة على استرجاع المعطيات ومكاملتها.
٣. Transformers لاستخدام النماذج التي توفرها منصة HuggingFace الشهيرة.
٤. Jinja2 مكتبة لبناء القوالب النصية، تم استخدامها لبناء قوالب الأوامر في نظام التقييم.
٥. Django & Django Rest Framework لبناء قسم النهاية الخلفية Backend لتطبيق المستخدم.
٦. React & Typescript لبناء واجهات تطبيق المستخدم.

إن تنجيز هذه المكونات ومكاملتها للحصول على هذه النظم يستلزم العمل بنهج متدرّج، حيث يتم تجزيء المتطلبات إلى سماتٍ أكثر بساطة والقيام بدورات مرحلية كاملة من أجل كل مكون. ولا تخلو كل دورة من وحدات الاختبار unit tests ونهاية كل مجموعة دورات من اختبارات المكاملة integration tests فلولاها لما تحققت أي من هذه النُظم.

٢.٤ الاختبارات والنتائج

تتطلب عملية الاختبار وجود مجموعة معطيات مرجعية لأسئلة وأجوبة مدعمة بالسياق، وعلى الرغم من شحة هذه المجموعات باللغة العربية، تمكننا من الحصول على مجموعة Arabic Reading Comprehension Dataset أو (ARCD)، التي تحتوي على 157 موضوعاً و1395 زوج سؤال–جواب مع سياق الإجابة الضروري لاختبار عملية الاسترجاع، كما وجدنا مجموعة ArabicaQA التي تحتوي على 89095 زوج سؤال–جواب مفتوح المجال، فيما يلي نتكلم عن الصعوبات التي تمت مواجهتها في عملية التطوير والتقييم ويليها النتائج التي حصلنا عليها على الرغم من هذه الصعوبات.

١.٢.٤ تحديات وصعوبات

واجهنا مجموعة من الصعوبات التي قد أثرت على المشروع على الرغم من محاولات حلها. وهي:

١. الصعوبة الأولى التي حدّت بشكل كبير من خيارات التطوير وجودة النتائج هي العتاد، فالعتاد السحابي الذي توفره منصة Colab، على الرغم من كونه أفضل خيار متوفر، لا تكفي لتشغيل النماذج اللغوية المتوسطة وقد تم حصر مجال الاختيارات الممكنة على النماذج اللغوية الصغيرة التي عدد متغيراتها أقل من 4 مليار متغير، التي

لا تمتلك إمكانيات استدلال قوية للتعامل مع سياق طويل (عدد ملفات مسترجعة أكبر) أو فهم النصوص الاختصاصية ضمن مجال معين.

٢. الصعوبة الثانية هي زمن التشغيل، حيث مقابل توفير بيئة Colab لبيئة عتادية بسعة 15GB GPU T4 إلا أنه يتحكم بوقت تشغيل البيئة على حسب الاستهلاك، وكان زمن التشغيل المعطى ضمن بيئة العمل 4 ساعات. بينما يستغرق توليد جواب لاختبار واحد دقيقة واحدة تقريباً مما يجعل عدد الاختبارات الممكنة 240 اختباراً فقط!

٢.٢.٤ تقييمات ونتائج

تتوفر العديد من أطر العمل لتقييم النظم المعتمدة على النماذج اللغوية ومن الأطر التي يمكن الوصول لها إطار العمل RAGAs أو Retrieval Augmented Generation Assessment الذي يتميز بمراعاة مقاييسه لآخر الأبحاث في مجال تقييم النماذج اللغوية. وبعد عدة محاولات لاستخدامه لم تؤتِ المحاولات بنتيجة لعدة أسباب:

١. توجد مشاكل برمجية في استخدام النماذج المتوفرة على منصة HuggingFace.

٢. تعتمد المكتبة تقييم الاختبارات على التوازي والذي يسبب مشاكل أثناء التشغيل على منصة Colab.

وهذا ما دفعنا للبحث عن طرق تقييم أفضل باستخدام منهجية "النموذج كحكم" أو LLM-as-Judge بالإضافة للتقييم باستخدام مقياس تشابه الإجابة الدلالي SAS.

تم تقييم النظام على 300 زوج سؤال–جواب من مجموعة ARCD، بعد فهرسة كامل محتوى المواضيع البالغ عدده 155 موضوعاً مختلفاً. وباستخدام نموذج تضمين mhaseeb1604/bge-m3-law المدرب على نصوص لغة عربية للمجال المالي لمهمة تشابه النصوص، ونموذج توليد Qwen/Qwen2.5-1.5B-Instruct مفتوح المصدر حصلنا على تقييم SAS **52.8%** وتم تقييم أداء مكون الاسترجاع في هذه الحالة بمقاييس Recall للحصول على نتيجة **90%**.

أعدنا الاختبار بإجراء تحسينات على نماذج التضمين والتوليد واختيارها من لوائح صدارة على HuggingFace حصلنا على تقييم SAS **65.9%** باستخدام نموذج تضمين sayed0am/arabic-english-bge-m3 ونموذج

توليد microsoft/Phi-4-mini-instruct، وتابعنا التقييم باستخدام مقاييس "النموذج كمحكم" لنحصل على التقييمات التالية:

CP(1-5)	CR(1-5)	RR(1-5)	F(1-5)	FC(1-5)
3.6	4	4.0	3.9	3.8

رابط الوصول إلى المشروع:

https://github.com/mekdad057/generic-domain-RAG-based-chatbot