Book '**Developing Apps with GPT-4 and ChatGPT**' , By Olivier Caelen, Marie-Alice Blete

Post 1 - Before LLM

محتاجين قبل ما نعرف شويه عن ال LLM وازاى ال models ديه بتعمل generate لل text ، هو إننا ناخد خطوة ونشوف الموضوع بدء إزاى من عند ال n-grams ، وازاى كان بناء على ال corpus الى عندى وال frequence بتاعت ال n-grams فقدر انى احدد انى الكلمة ديه هى المتوقع تيجي فى ال next predicted token فى الجملة ، وده لانها ظهرت ب frequency اكبر من الكلمات الاخر بإنها بتيجي مع الكلمتين او التلاته او ال n-grams دول ، مثلا زى

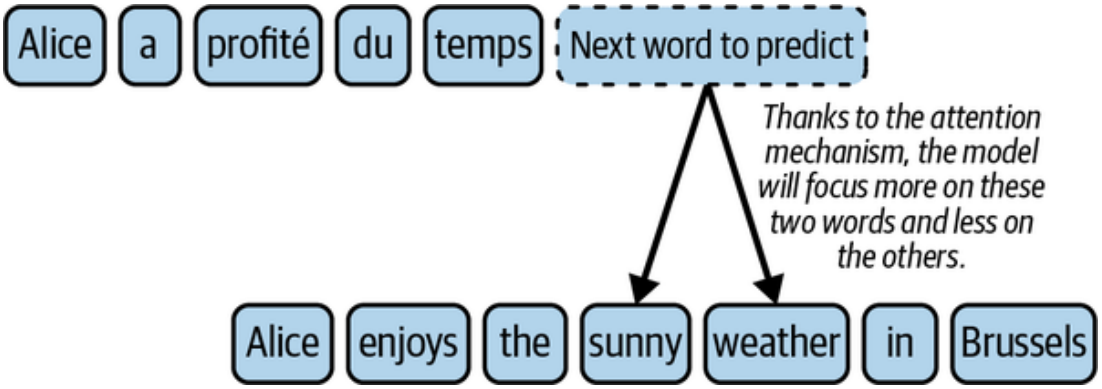
عمر بن ، هنلاقى المتوقع الخطاب ، بس ماذا لو كان سياق الكلام بيكلم عن عمر بن عبدالعزيز ؟ هنا بقا احنا ملناش دعوه بالسياق لاننا معتمدين على ال frequency بتاع ظهور كلمة الخطاب مع عمر بن ، اكتر من ظهورها مع عبدالعزيز رضى الله عنهما وعن سائر أصحاب رسولنا الكريم صل الله عليه وسلم .

جميل ابتدينا اننا ناخد خطوة اخرى نحاول منها اكتر نفهم ال context وكمان ال grammar بتاع الجمل وهنا ظهرت بعد فترى ال RNN and LSTM والى قعدت مستخدمه فترة كبيرة بسبب انها كانت بتدى نتايج كويسة ، وخاصه ظهرت لموضوع الترجمات ، ولكن كان فيه مشكلة اخرى وانى النوع ده من ال network كان بياخد وقت كبير جدا ومينفعش انك تعمله process in parallel بسبب انى كل time step معتمده على ال output بتاع الى قبلها وكمان ال complexity بتبقا اكبر كل ما عدد ال tokens بقا اكبر ، فلو عندك text فيه token 5000 كانك عند netrok فيها layer 5000 !!!

وكمان مع حجم ال data الى بنشوف ال LLM متدرب عليها كان هيبقا صعب جدا مع ال models ديه .

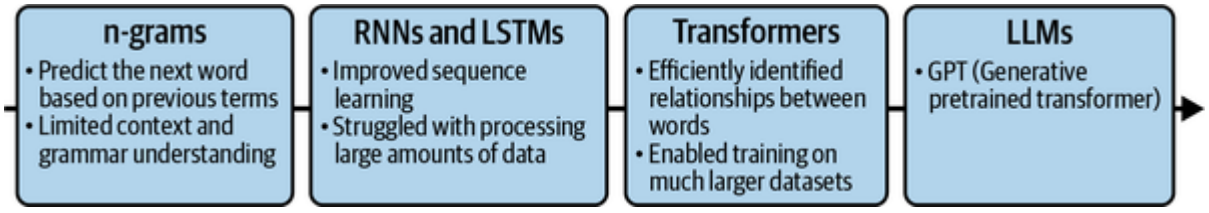
ولكن جه بعد كده فى 2017 ال paper بتاعت transformers all you need والى غيرت الدنيا فى ال nlp ، لانه عالج اولا مشكلة ال long text ، وكمان ال parallelization ومشاكل ال vanishing and exploding

problems الى كانت فى ال RNN and LSTM الى بتحصل بسبب عدد ال tokens .
وهنا ال transformers فيه ما يسمى بال attention mechanism والى فيه انى مش بدى اهميه لكل الكلمات الى فى ال text بنفس ال weights لا انا ممكن يكون ال predicted token الجاى معتمد على كلمتين فقط ومش لازم يكونوا كمان الكلمتين دول ورا بعض ، فهنا ال prediction بتاع ال token هيكون معتمد على الكلمات المهمه ليه والكلمات الاخرى هتاخد weights اقل ، زى ما فى المثال :



هنا مثلا الكلمة الى انا هتوقعها فى ال next time step هى ensoleillé والى هى معناها sunny فالبنسبة ليها اكتر كلمتين هى محتاجهم هما sunny and weather .

بعد كده ننتقل لنقطة ال GPUs وال Parallelization وهى انى ال transformers models قادره انها تهندل ال text بتاعتك بشكل simuntuiosly مش sequentially وده متناسب مع ال GPUs ويخليك تقدر تعمل train على داتا كبيره فى وقت اقل.



Post 2 - LLM

فى 2017 اتنشرت attention is all you need والى كان فيه ال transformers مع فكرة ال attention ، وهنا فى ال transformers فى جزئين مهمين هما ال encoder and decoder بعض ال models هى فقط encoder والى بيكون دورة انه يطلع ليك numerical representation لل text بتاعك ، والبعض الاخر وهو ال decoder والى بيكون دورة هو انه يحاول يعمل generate ل text بناء على ال encoder output وحاجات اخرى ، وفى بعض ال models بتضمن الاتنين ال encoder and decoder .

طب هنا بالنسبة لل gpt مثلا فهو decoder فقط وهنا ده بيعتمد على نوع من ال attention وهو ال self attention عشان يقدر يعمل genrate لل text .

طيب دلوقت بقا بالنسبة ل LLM زى GPT الى هو شغال ك decoder فقط كان بيروح ليها prompt الى هو كانه ال start بتاع الكلام وهو يبتدى يكمل ليك الكلام ويعمل genrate لباقى ال sequence زى

The weather is nice today, so I decided to

هتلاقى هو بناء على ال prompt input ديه هيروح يعمل gernrate مثلا ل :

go for a walk

وده يعتبر ما يسمى بال completion .

طب ازاى بقا ال LLM عمل ده ، فى البداية ال prompt الى انتا بعتهاله ديه بتروح يحصلها ما يسمى tokenization عشان فى الاخر تكون عباره عن list of tokens وهنا طبعا ال tokenization فيه انواع مختلفه ولكن نقدر نقول بشكل عام هو بيكون عباره عن اما single words or sub-words

فمثلا ممكن كلمه today تبقا عباره عن or punctuation and others
to and day الى هو two tokens .

ممكن تجرب ال tokenization بتاع open ai من هنا:
https://platform.openai.com/tokenizer
بعد كده بيكون بناء على ال tokens ديه يبتدى يفهم ال transformers
models with attention mechanism انهى اكتر tokens هتكون
مناسبه انها تروح ت predict ال next token وبعدين يضم ال token ده
لل prompt او ال input الى دخلته وبعدين يشوف ال token الى بعده
ويضمه وهكذا لحد ما يوصل لل max length بتاع ال models او انه يكون
انتا محدد .

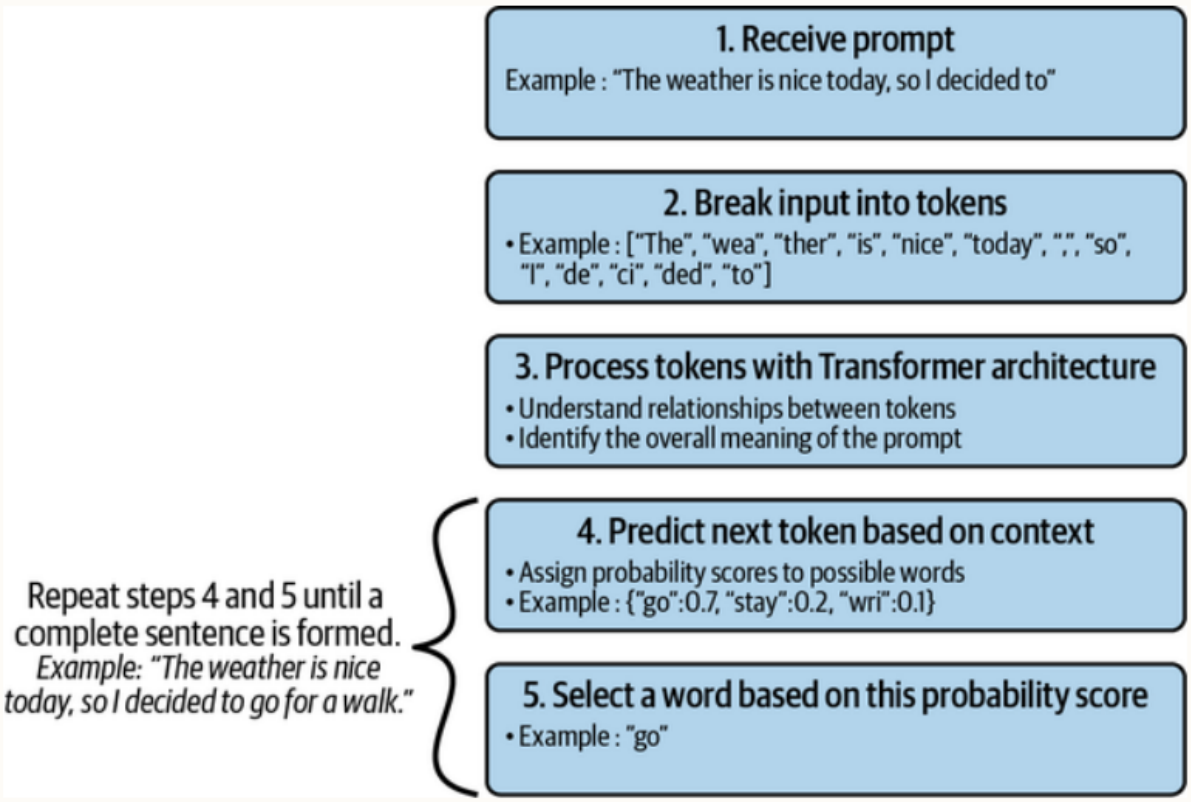


Figure 1-5. The completion process is iterative, token by token

Post 3 - History GPT-1 to GPT-4

قبل ما نكلم على GPT-1 محتاجين نعرف انى كان معظم ال NLP tasks معتمده بشكل او باخر على ال labeled data ، عشان تقدر تعمل train ل model على ال task الى انتا عايزه زى ال sentiment classification ، وده بيستهلك وقت كبير جدا وفلوس كتيره عشان تقدر تبنى dataset معتمده على ال manual classification الى من خلاله هتقدر انك تبنى ال model الى انتا عايزه .

فجه هنا بعد سنه من ال attention is all you need ، اتنشرت ال paper الخاصه ب GPT-1 من open ai ، والى كان فيها خطوة قبل عمليه ال nlp model الى هو مبنى على ال task بتاعك ، وهو إنك الأول ممكن تبنى pretrained model يقدر يتوقع ال next token من داتا كبيرة جدا ، وبعدين تاخد ال model ده تعمله fine tune على ال nlp task الى عندك فى ال classification .

بالطريقه ديه ال model الاول ده عرف معلومات كتيره عن اللغة عرف ال context بتاع اللغة وال grammer بتاع اللغة وغيره وعلى حسب حجم الداتا برضه ال model بيقدر يكون افضل معاك فى ال downstream task ال عندك ، وبكده تقدر تاخد ال model ده مش بس ل sentiment problem لا تقدر تاخده لاكتر من task .

لكن هنا GPT-1 كان معمولة train على حجم داتا مش كبير فكان يشتغل كويس مع ال completion task الى حد ما ولكن هيحتاج انك تعمله fine tune لو عايز مثلا تستخدمه فى classifcation problem .

ولكن فى النهايه بتحتاج labeled data صغيره عشان تقدر توصل لنتايج كويسه لل task بتاعتك بدل ما كنا بنحتاج ل massive label data عشان ال model يدرب عليها .

بعد كده حه ال GPT-2 فى 2019 والى كان عباره عن scalable model من GPT-1 معمولة train على داتا اكبر حوالى 40 جيجا وليه كمان parameters اكتر 1.5 billion وده كان افضل من ناحية ال completement لل text وكمان افضل انك تاخده على down stream task ، بسبب تنوع وزياده ال data لان GPT-1 كان معمولة train على 11 الف كتاب ، فهنا ممكن يكون ال sequence فى الكتب مختلف عن ال social media مختلف عن ال web articles ، مختلف عن ال scientific articles or questions and answers regarding of any domain .
وده الى هيتم معالجته فى ال new GPTs .

Post 4 - History GPT-1 to GPT-4

نيجبى بعد كده لل GPT-3 فى 2020 والنقلة النوعيه فى حجم ال model نفسه والى كان من 1.5 فى GPT-2 ل 175 Billon paramter فى GPT-3 ، وكمان حجم ال data الى حصل عليها train كان كبير جدا ومن مصادر مختلفه فمثلا billions of webpages من :
https://commoncrawl.org/

ومن Wikipedia ومن كتب وغيره من ال resources ، هنا ال GPT-3 قدر انه يفهم complex and deep pattern فى ال text مع اختلاف المصادر والكتب والكتاب وغيره من ال data الى حصله train عليها قدر يفهم طريقه الكتابه وبقا عنده فهم عميق للغة ، وكمان بقا يعمل generate لنص متناسق وموزون تقدر تعتبره كان حد الى كاتب الكلام مش AI model ، مش بس كده لانه كمان اتعمله train على حجم ضخم من ال web pages بقا قادر انه generate code ، وكمان ممكن ميحتجش انك تعمل fine tune ويعمل هو ال task الى انتا كنت عايز تعملها fine tune .

ولكن هنا كان على حسب ال data برضه ال model معموله train عليها فممكن انك تلاقى ال model مطلع محتوى سيء من نواحى مختلفه ، كمان ممكن يكون فيه عدم تناسق بين ال end user عايزه والى ال model حصله train عليها بذات انه متعلم ازاى يعمل generate لل next token وده مش الهدف الاخير الى انا عايز ال model عشانه ، وهنا ظهر model اخر لل GPT-3 وهو ال Instruct GPT-3 فى 2021 وده كان معموله optimization من خلال ال reinforcement learning من ال human feedback ، ببساطة انا هخلى ال model يتعمل بناء على ال instructions بتاعت ال feed back الى جاى منى فيقدر يخلى ال model بيتعلم طول الوقت .

فهنا بقا مثلا هنيجي نقول لل model :

Explain what is meant by time complexity

مع GPT-3 لو اديته ال prompt ديه هيروح يكمل بالشكل ده مثلا يعنى :

Explain what is meant by space complexity. Explain what is meant by the big-O notation.

لكن مع With the instructGPT-3 :

Time complexity is a way of measuring the amount of time it takes for an algorithm to run and complete its task. It is usually expressed using Big O notation, which measures the complexity of an algorithm in terms of the number of operations it performs. The time complexity of an algorithm is important because it determines how efficient the algorithm is and how well it scales with larger inputs.

فهنا هو فهم انا محتاج ايه ومرحش كمل زى ما فى ال GPT-3 .

طب ازاى ده حصل ؟

Post 5 *From GPT-3 to InstructGPT*

عشان نروح من GPT-3 ل InstructGPT فيه خطوتين مهمين وهما supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF)

ال SFT هنا بتاخد ال model الى هو GPT-3 وتبتدى انها تعمل fine tune ولكن ب supervised من human فمثلا ناخد random prompt من ال dataset الى فيها prompt او اسئلة مثلا عندى ، واروح اخلى اكتر من انسان يرد على نفس ال prompt ديه ، فهنا بقا عندى السؤال الواحد ليه اكتر من إجابه على حسب معلومات كل بنى ادم .

هنا هيتكون عندى مجموعه من ال dataset الى بقا ليها prompt + answer from human الى هو دلوقت قدر اخد GPT-3 واروح اعمله fine tune بإستخدام ال data الى اتكونت ديه.

هنا ال model ده هو ما يسمى بال SFT .

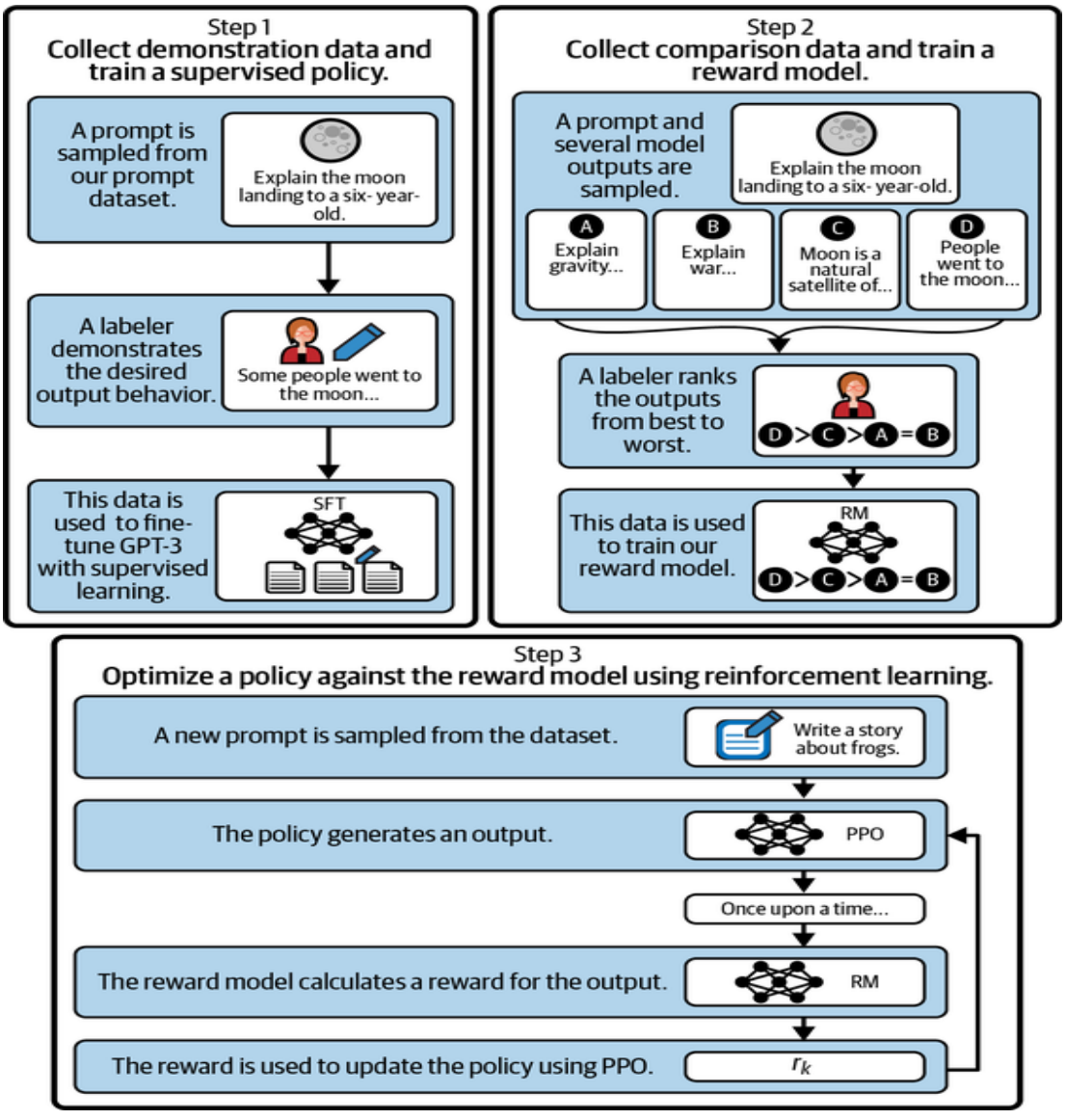
بال RLHF متقسمه لخطوتين الأولى :

اروح اعمل random prompt من الى عندى ، واخلى ال model هو SFT يطلع اكتر من إجابه بناء على ال prompt ديه لانه كان مدرب من خلال انى نفس ال prompt اتجاوب عليها اكتر من مره من اشخاص مختلفين ، ويبتدى بعد مال model يطلع الاجابات المختلفه ، يبتدى برضه حد عنده خبره يعمل score للإجابات المختلفه ديه ويرتبها على حسب الأفضلية ، بعد كده بقا هناخد ال dataset الى اتكونت ديه والى بقا فيها كل prompt والإجابة بتاعتها ليهم score معين عشان ابتدى ادرب ما يسمى ب rewarded model الى هيكون عباره عن انى اخد ال SFT model عشان ادربه على ال scored data ديه فيدينى ما يسمى بال Rewarded model .

الهدف من ال Rewarded model هو انى قدر اعمل score دلوقت للاجابه الى ال model هيطلعها ، بمعنى انى بناء على ال answer الى ال model هيدهانى هتكون متناسبه مع ال prompt ولا لا هيكون ال score ده اما عالى او واطى .

الخطوه التانيه بقا هى انى اروح اخد ال Rewarded model ده واستخدم معا Reinforcement Learning عشان يبقا عندى ما يسمى بال InstructGPT-3 .

هنا بقا هنبتدى ناخد random prompt من اى user واخلى مثلا ال GPT model او ال SFT model يجاوب على ال Prompt ديه ، بعدين بناء على ال output ده هروح بال Rewarded Model ادى score لل output الى model طلعه واعمله evaluation ، بناء على ال rewarded الى ال model استلمها يبتدى يعمل update لنفسه وهنا انتقلنا انى يكون الموضوع automated مش معتمد على human بقا معتمد على انى ال model يعمل generate و model تانى يقيم ال generation ده و model تالت يحصله update بناء على التقيم الى استلمه.



Post 6 *GPT-3.5, Codex, and ChatGPT*

بعد كده فى 2022 open ai عرفت GPT-3.5 and codex and chatgpt ، وديه كان عباره عن new versions من ال GPT-3 تندرج تحت GPT-3.5 ، ال codex هو LLM ولكنه معموله train على billions of lines of codes ، وهو المعتمد عليه فى Github copilot ، ولكن codex بقا depracted وبقا المستخدم حاليا هو GPT-3.5 turbo and GPT4 .
بعدين بقا فيه Copilot X الى بقا معتمد على GPT-4 .
وفى الغالب chatgpt ال base model بتاعه هو ال GPT-3.5 ، والى هو ظهر فى اواخر 2022 وعمل ضجه كبيره فى عالم ال NLP وفى ال Business بشكل عام ، وفى مجالات كتير ، وبقينا حتى بنستخدمه عشان نتعلم من خلاله ك interactive teacher ليك انتا لوحدك .

فى 2023 ابتدى ظهور GPT-4 والى هو كان بيدى نتايج افضل واكثر دقه من chatgpt على رغم من انى chatgpt بيدى نتايج كويسه الا ان GPT-4 افضل وكمان فيه انك تقدر تستخدم معاه الصور وتقدر تعرف الى فى الصورة او تسال اسئله عن ايه الى فى الصورة بظبط .
وكمان فى بعض ال exams مقارنه بين chatgpt and gpt-4 لقو انى chatgpt جايب 10% فى واحد و 31% فى التانى بينما GPT-4 جايب 90% و 99% !

Table 1-1. Evolution of the GPT models

2017	The paper “Attention Is All You Need” by Vaswani et al. is published.
2018	The first GPT model is introduced with 117 million parameters.
2019	The GPT-2 model is introduced with 1.5 billion parameters.
2020	The GPT-3 model is introduced with 175 billion parameters.
2022	The GPT-3.5 (ChatGPT) model is introduced with 175 billion parameters.
2023	The GPT-4 model is introduced, but the number of parameters is not disclosed.

فى اخر ال chapters الكاتب اكلم عن ال uses cases المستخدمه حاليا من الشركات مبنية على GPT-4 و chatgpt منها:

Be My Eyes

Morgan Stanley

Khan Academy

Duolingo

Yabble

Waymark