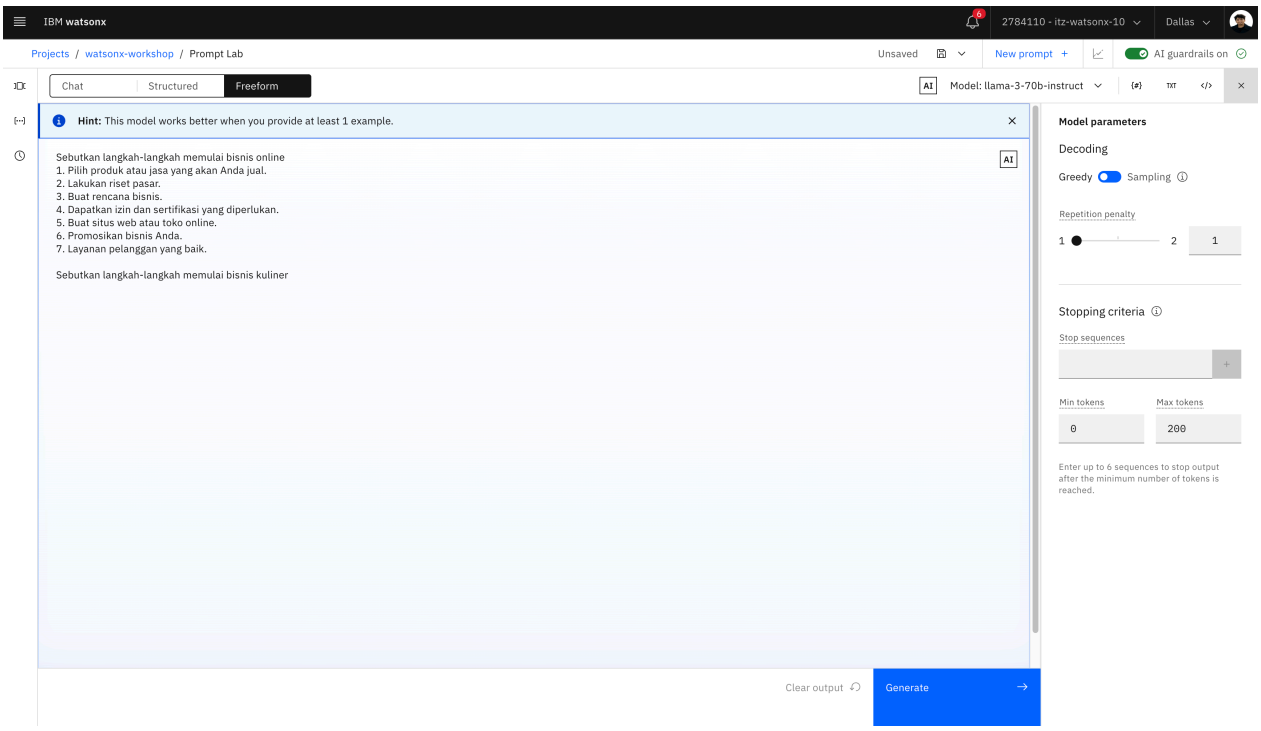
Prompt engineering

1.0 LLM Foundations

Sebelum kita mulai mengeksplorasi kemampuan watsonx.ai, pertama-tama kita perlu mengetahui tentang cara kerja Model Bahasa Besar (LLM), dan bagaimana kita dapat menyesuaikan model dan parameter untuk mengubah output yang dihasilkan. Mengetahui hal tersebut dapat membantu kita untuk lebih efisien menggunakan LLM model.



Saat Anda membuka watsonx.ai, ini adalah tampilan yang akan ditunjukkan kepada Anda. Area teks tengah yang besar disebut Prompt Lab, atau Prompt Builder jika Anda memilih tampilan lebih lanjut dengan mengklik kotak centang di kiri atas. Di sisi kanan adalah parameter dari model yang dapat Anda gunakan untuk memilih guna mengoptimalkan respons model terhadap permintaan atau prompt Anda. Dan di kiri bawah, terdapat ringkasan jumlah token yang digunakan oleh prompt Anda selama eksekusi.

1.1 Tokens

Each time you enter a prompt, your “input tokens” and “generated tokens” will update. Tokens are an important concept to understand as they constrain the performance of your model plus determine the cost of using models. As you will learn throughout the Labs, tokens are not a 1:1 match with words in natural language, but on average, one token is equal to 4 characters. Before sending your prompt to the model, the prompt's text is Tokenized or broken into smaller subsets of characters better understood by a model.

It is important to monitor your token usage to know how much information you are feeding into the model with each prompt, as well as how much text is generated for you. Depending on the model selected in Prompt Builder, you will see a max of 2048 or 4096 tokens. Keep in mind that the more expressive you are with your prompt instructions, the less room the model has to respond back to you.

Setiap kali Anda melakukan prompting, “input tokens” dan “generated tokens” akan diperbarui. Token adalah konsep penting untuk dipahami karena membatasi performa model Anda dan menentukan biaya penggunaan model. Seperti yang akan Anda pelajari di melalui Lab, tokens bukanlah 1:1 dengan kata-kata yang ada dalam sebuah kalimat, namun rata-rata, satu token sama dengan 4 karakter. Sebelum mengirimkan perintah Anda ke model, teks perintah tersebut di-Tokenized atau dipecah menjadi subkumpulan karakter yang lebih kecil yang lebih mudah dipahami oleh model.

Penting untuk memantau penggunaan token Anda untuk mengetahui berapa banyak informasi yang Anda masukkan ke dalam model dalam setiap perintah, serta berapa banyak teks yang dihasilkan untuk Anda. Bergantung pada model yang dipilih di Prompt Builder, Anda akan melihat maks 2048 atau 4096 token. Ingatlah bahwa semakin ekspresif Anda dalam memberikan instruksi, semakin sedikit "ruang" yang dimiliki model untuk merespons Anda.

1.2 Everything is text completion

watsonx.ai bukan chatbot interface, terkadang hasil prompt tidak memberikan apa yang diharapkan, tapi kita dapat memberikan instruksi ke LLM model untuk melakukan sejumlah hal dengan cara yang lebih tepat. Misalnya, bagaimana jika kita meminta Watsonx.ai untuk: Sebutkan langkah-langkah memulai bisnis online

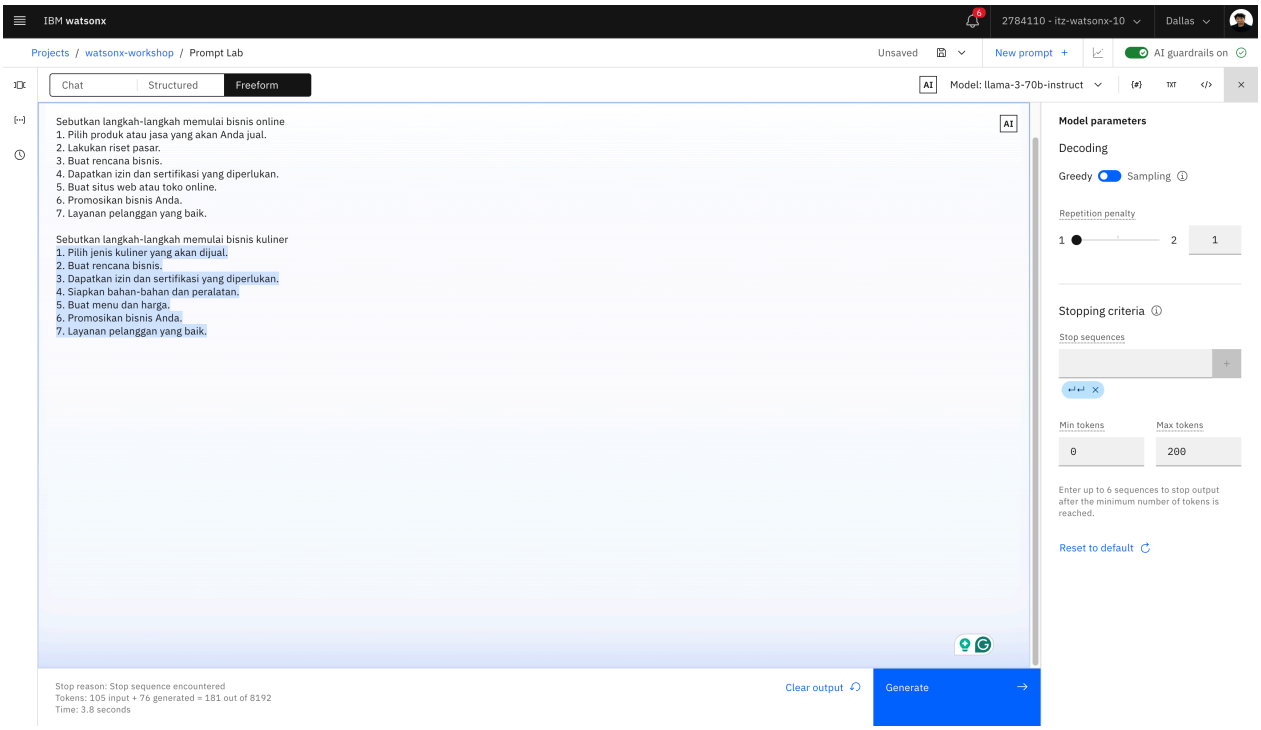
Sebutkan langkah-langkah memulai bisnis online

Daftarkan diri Anda untuk menjadi seorang pebisnis online.

Jawaban di atas bukanlah apa yang kita harapkan.

1.3 Provide an example as guidance (or Single Shot Prompting)

Untuk mendapatkan respons dengan kualitas yang lebih baik, berikan contoh jenis respons yang Anda inginkan. Dalam istilah teknis, ini disebut Single Shot Prompting.



Note: Gambar berikut menunjukkan hasil dari watsonx.ai. Teks berwarna abu-abu adalah contoh input yang dapat berikan untuk model. Teks yang di-highlight biru adalah respons dari model.

Seperti yang Anda lihat, memberikan satu contoh sebelum memberikan perintah ke LLM disebut dengan Single Shot Prompting, namun menambahkan lebih banyak contoh ke dalam prompt juga lumrah untuk dilakukan. Umumnya, memberikan contoh dalam jumlah yang lebih dari satu disebut sebagai “Few Shot Prompting” dan merupakan cara yang ampuh untuk memastikan Anda mendapatkan hasil yang spesifik.

Gunakan teks berikut ini, untuk mencoba Prompt Builder:

```
Sebutkan langkah-langkah memulai bisnis online
1. Pilih produk atau jasa yang akan Anda jual.
2. Lakukan riset pasar.
3. Buat rencana bisnis.
4. Dapatkan izin dan sertifikasi yang diperlukan.
5. Buat situs web atau toko online.
```

```
6. Promosikan bisnis Anda.
7. Layanan pelanggan yang baik.

Sebutkan langkah-langkah memulai bisnis kuliner
```

1.4 Include descriptive details

Semakin detail instruksi yang diberikan, semakin baik:

- Content
- Style
- Length dari response yang diberikan

Sebutkan langkah-langkah memulai bisnis online
1. Pilih produk atau jasa yang akan Anda jual.
2. Lakukan riset pasar.
3. Buat rencana bisnis.
4. Dapatkan izin dan sertifikasi yang diperlukan.
5. Buat situs web atau toko online.
6. Promosikan bisnis Anda.
7. Layanan pelanggan yang baik.

Berikan 5 langkah-langkah untuk memulai bisnis kuliner dalam skala besar

1. Pilih konsep bisnis kuliner yang akan Anda jual.
2. Lakukan riset pasar untuk mengetahui apa saja yang diperlukan.
3. Buat rencana bisnis yang mencakup segala sesuatu, dari pengadaan bahan-bahan hingga pemasaran.
4. Dapatkan izin dan sertifikasi yang diperlukan untuk memulai bisnis kuliner.
5. Buat sistem manajemen persediaan dan pengiriman makanan yang efektif.

Model Parameter

2.0 Adjusting Model Behavior

Perubahan pertama yang bisa kita lakukan adalah model (LLM) apa yang kita gunakan untuk mengeksekusi prompt kita. Ini adalah salah satu perubahan terbesar yang dapat Anda lakukan, karena model tertentu dibuat lebih baik untuk tugas tertentu. Latihan selanjutnya di lab ini akan meminta Anda mengubah model yang digunakan jika ingin menjawab beberapa pertanyaan yang lebih menantang.

Secara umum, beberapa model bekerja lebih baik dengan peringkasan, kata kunci, dan semantik, sementara model lainnya bekerja lebih baik dengan teks terstruktur seperti HTML, markdown, atau JSON. Cara terbaik untuk mengetahui model mana yang cocok untuk kasus penggunaan Anda adalah dengan mengujinya, namun penting untuk mengetahui bahwa pilihan model dapat membuat perbedaan besar!

watsonx.ai juga menyediakan beberapa parameter untuk mengonfigurasi cara LLM merespons permintaan. Memilih parameter yang benar seringkali lebih merupakan seni daripada sains. Menginvestasikan waktu untuk memahami prompting dan kemudian mengubah parameter dari model dapat membantu menghasilkan respons yang lebih baik.

Coba berkesperimen dalam pengaturan parameter dengan menggunakan teks berikut:

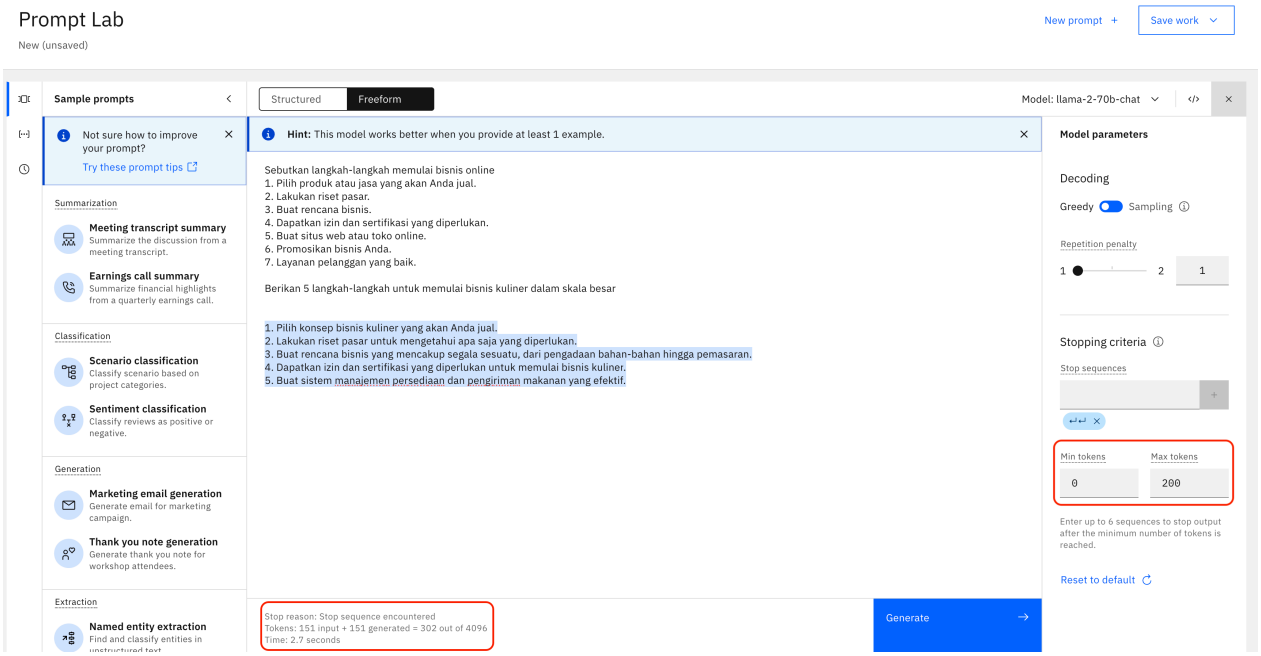
```
Sebutkan langkah-langkah memulai bisnis online
1. Pilih produk atau jasa yang akan Anda jual.
2. Lakukan riset pasar.
3. Buat rencana bisnis.
4. Dapatkan izin dan sertifikasi yang diperlukan.
5. Buat situs web atau toko online.
6. Promosikan bisnis Anda.
7. Layanan pelanggan yang baik.

Sebutkan langkah-langkah memulai bisnis kuliner
```

2.1 Set the min and max tokens

Jika Anda merasa teks yang dihasilkan terlalu pendek atau panjang, coba sesuaikan parameter yang mengontrol jumlah token baru:

- Parameter **Min new tokens** mengontrol jumlah minimum token (~kata) dari respons yang dihasilkan
- Parameter **Max new tokens** mengontrol jumlah maksimum token (~kata) dari respons yang dihasilkan



2.2 Specify stop sequences

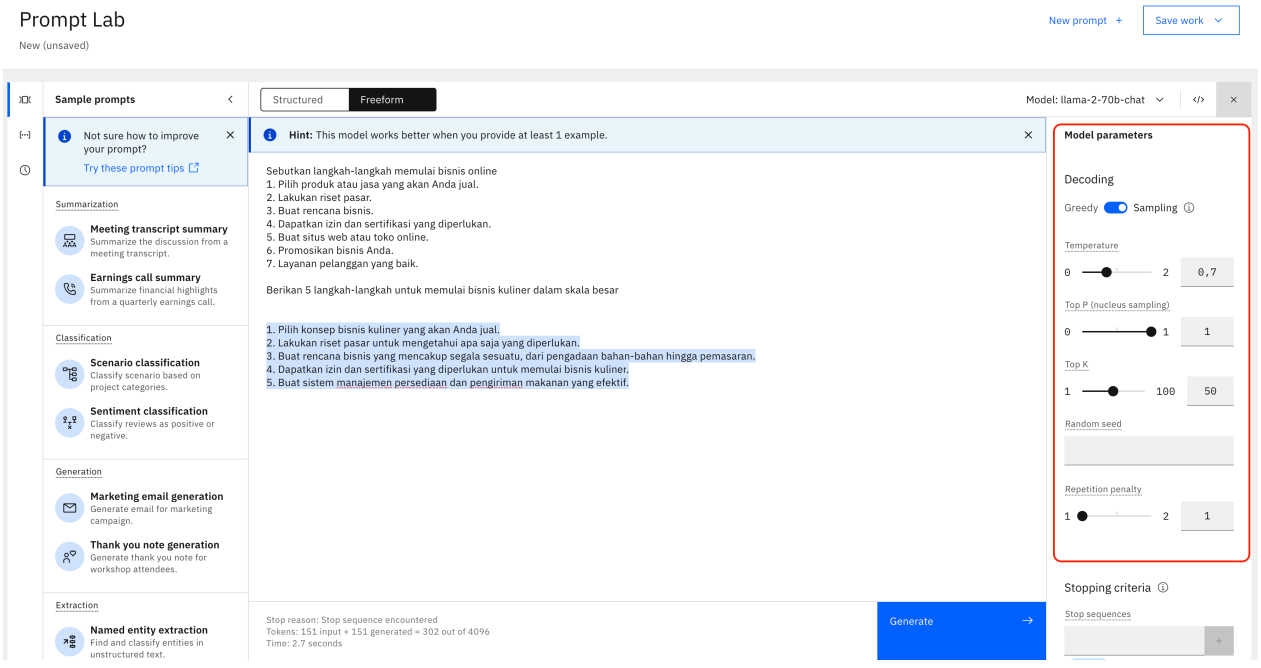
Jika Anda menentukan *stop sequence*, output akan otomatis berhenti ketika salah satu *stop sequence* muncul pada keluaran yang dihasilkan. **double enter** biasa ditambahkan mencegah model mengenerate ulang output dikarenakan jumlah token yang dihasilkan belum mencapai *max tokens*

2.3 Adjust decoding parameters

Jika responsnya terlalu umum atau menyimpang, pertimbangkan untuk mengatur parameter decoding. Atau ketika response mungkin kurang kreatif, pengaturan juga ada baiknya dilakukan.

`Decoding adalah proses menentukan urutan keluaran berdasarkan urutan masukan`

- `Greedy decoding` memilih kata dengan probabilitas tertinggi pada setiap langkah proses decoding.
- `Sampling decoding` memilih kata-kata dari distribusi probabilitas di setiap langkah
- `Temperatur` mengacu pada pemilihan kata dengan probabilitas tinggi atau rendah. Nilai temperature yang lebih tinggi menyebabkan lebih banyak variabilitas.
- Top-p (pengambilan sampel inti) mengacu pada pemilihan kumpulan kata terkecil yang probabilitas kumulatifnya melebihi p.
- Top-k mengacu pada pemilihan k-kata dengan probabilitas tertinggi di setiap langkah. Nilai yang lebih tinggi menyebabkan lebih banyak variabilitas. Keuntungan dari penguraian Greedy decosing adalah Anda akan melihat hasil yang dapat direproduksi. Ini dapat berguna saat melakukan pengetesan. Menyetel temperature=0 dalam pendekatan sampling decoding memberikan variasi yang sama seperti Greedy decoding.



2.4 Add a repetition penalty

Terkadang, Anda akan melihat teks yang diulang-ulang. Menaikkan temperature terkadang dapat menyelesaikan masalah. Namun, ketika teks masih berulang bahkan dengan dengan pengaturan temperature yang lebih tinggi, Anda dapat mencoba menambahkan *repetition penality*. Semakin tinggi nilai penalty-nya, semakin kecil kemungkinan terdapatnya teks berulang.

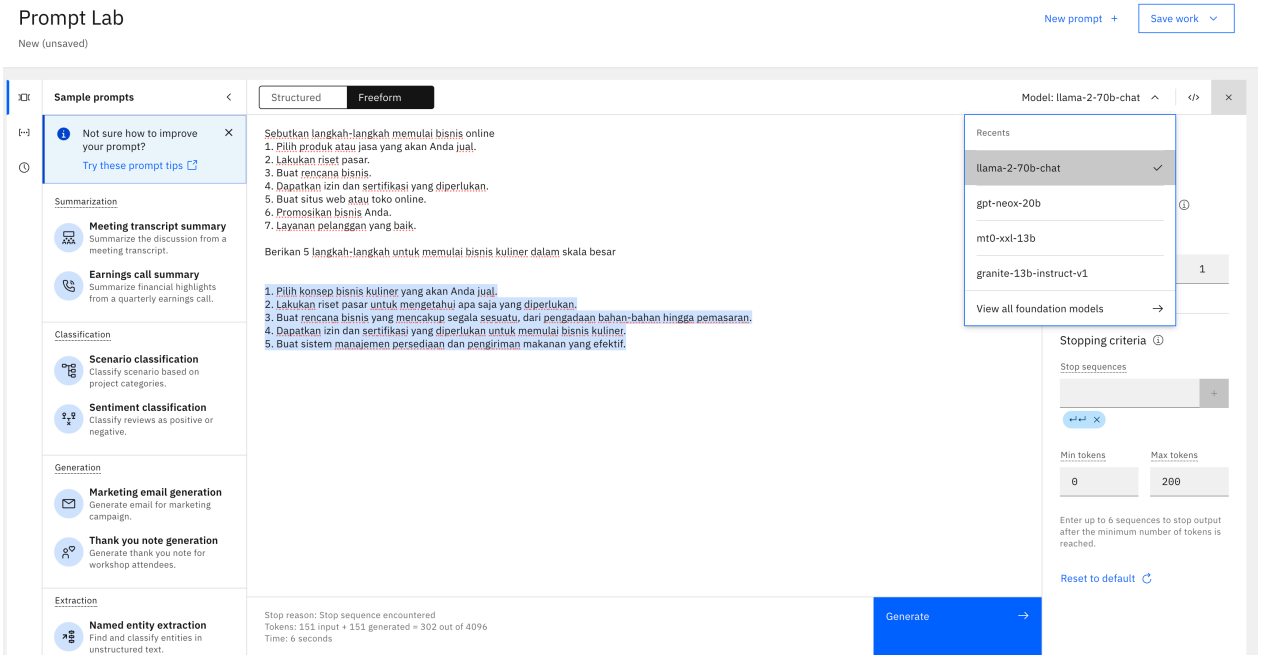
2.5 Excellent 3rd party blog post on model parameters

Uraian di atas memberikan pengenalan yang cukup baik tentang parameter apa saja yang model. Namun Anda harus membaca artikel ini tentang parameter yang dimiliki model karena dapat memberikan contoh tambahan yang sangat baik tentang bagaimana parameter model bekerja ditambah ada ilustrasi yang dapat membantu Anda lebih memahami konsepnya. Semakin baik Anda memahami parameter model, semakin anda terhindar dari frustrasi dan semakin mudah untuk menyesuaikan model agar berfungsi sesuai kebutuhan Anda.

General advice

3.1 Try different models

Jika prompt anda tidak menghasilkan apa yang anda inginkan, terutama apabila prompt dibuat dalam Bahasa Indonesia, cobalah ganti pilihan model yang digunakan



3.2 Check your use case

LLM memiliki potensi besar, tetapi LLM tidak memiliki logika, pengetahuan, dan domain expert. Beberapa kasus lebih cocok dibandingkan dengan kasus yang lain: LLM unggul dalam tugas-tugas yang melibatkan *text generation* atau mengenali kesamaan pattern dan mentransformasikan input teks yang diberikan.

Jika *prompt* yang diberikan sudah menggunakan *best pracatice* yang dibahas di sini, namun tetap tidak mendapatkan hasil yang baik dari model mana pun, mungkin saja kasus yang dipilih merupakan hal yang tidak dapat ditangani dengan baik oleh LLM.

Misalnya, meskipun kita bisa mendapatkan hasil yang cukup layak untuk aritmatika sederhana, LLM umumnya tidak bisa mengerjakan matematika dengan baik: Researchers find that large language models struggle with math

4.0 Balancing intelligence and security

Dengan kecerdasan buatan yang hebat, terdapat risiko keamanan yang lebih tinggi. Solusi seperti ChatGPT adalah Very large Language Model (VLLM) dengan 175 miliar parameter. Mereka dibuat oleh tim OpenAI menggunakan kumpulan data Obrolan non-publik sebagai tambahan dan kumpulan data Reinforcement Learning Human Feedback (RLHF). ChatGPT adalah LLM yang dibuat seperti chatbot.

Di watsonx.ai, kami berinteraksi langsung dengan LLM yang lebih kecil (3-20 miliar parameter). Ini adalah pilihan bijak dari sudut pandang keamanan. *prompt injection* merupakan risiko besar bagi penggunaan LLM di perusahaan. Dengan menggunakan *prompt injection* , hacker akan membuat perintah rumit yang menyebabkan LLM seperti ChatGPT mengabaikan/melewati protokol keamanan dan mengungkapkan informasi sensitif perusahaan. Bayangkan saja Anda seorang hacker. Model manakah yang akan Anda pilih sebagai target *prompt injection* ? ChatGPT OpenAI dengan 175 miliar parameter yang mampu melakukan ribuan tugas atau model parameter 3 miliar yang lebih kecil dan lebih fokus sangat disesuaikan untuk beberapa tugas terisolasi? Manakah yang memiliki *attack surface* yang lebih luas untuk terjadinya *prompt re-engineering*?

Model yang lebih kecil dan sederhana di watsonx.ai lebih menantang bagi calon hacker. Menggunakan banyak model kecil dibandingkan satu model besar seperti ChatGPT menciptakan distribusi *sensitive entry points* yang lebih luas. Setiap model bahasa yang kecil jauh lebih sulit untuk dimanipulasi karena fungsinya yang terbatas dan *prompt engineering* tingkat tinggi yang diperlukan untuk melakukan tugas utamanya. Mereka tidak memiliki berbagai fungsi seperti ChatGPT. Seperti yang diketahui para programmer, kehilangan semua sumber daya Anda hanya dengan satu kegagalan adalah tindakan yang tidak bijaksana. Jauh lebih baik untuk menguraikan solusi Anda demi keamanan, skalabilitas, dan kontrol.

Untuk keamanan, lebih kecil lebih baik. Selain manfaat keamanan, terdapat peningkatan komputasi dengan menggunakan model yang lebih kecil dan berbobot lebih ringan. Mari kita lebih banyak berinteraksi dengan LLM watsonx.ai untuk lebih memahami dan mempelajari cara membuat mereka merespons sesuai kebutuhan kita.

Further Reading

- OpenAI prompt intro
- OpenAI prompt engineering tutorial
- co:here prompt engineering tutorial