ADEPT

Announcements Research

Fuyu-8B: A Multimodal Architecture for AI Agents

October 17, 2023 — Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, Sağnak Taşırlar

We're open-sourcing Fuyu-8B - a small version of the multimodal model that powers our product.

We're releasing Fuyu-8B, a small version of the multimodal[1] model that powers our product. The model is available on HuggingFace. We think Fuyu-8B is exciting because:

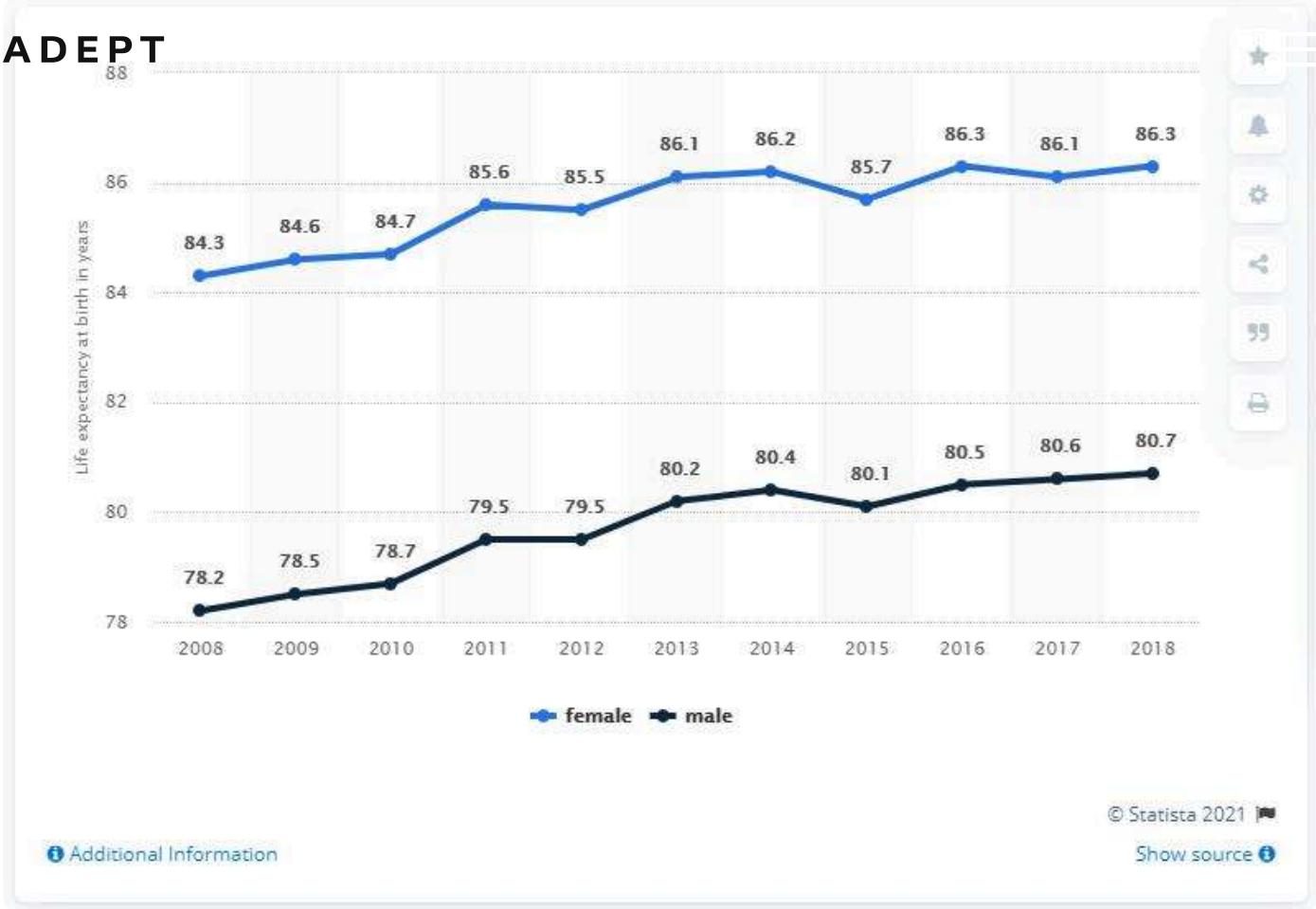
1. It has a much simpler architecture and training procedure than other multi-modal models, which makes it easier to understand, scale, and deploy.
2. It's designed from the ground up for digital agents, so it can support arbitrary image resolutions, answer questions about graphs and diagrams, answer UI-based questions, and do fine-grained

localization on screen images.

ADEPT

3. It's fast - we can get responses for large images in less than 100 milliseconds.
4. Despite being optimized for our use-case, it performs well at standard image understanding benchmarks such as visual question-answering and natural-image-captioning.

Fuyu's caption: "A cake with writing on it that says congratulations kate and luke on your upcoming arrival."



Question: "What is the highest life expectancy at birth of males?"

Fuyu's answer: "The life expectancy at birth of males in 2018 is 80.7"

Today, we're releasing Fuyu-8B with an open license (CC-BY-NC)—we're excited to see what the community builds on top of it! We also discuss results for Fuyu-Medium (a larger model we're not releasing) and provide a sneak peek of some capabilities that are exclusive to our internal models.

Because this is a raw model release, we have not added further instruction-tuning, postprocessing or sampling strategies to control for undesirable outputs. You should expect to have to fine-tune the model for your use-case.[2]

Model Architecture

Adept is building a generally intelligent copilot for knowledge workers. In order to do this, it's important for us to be able to understand user context and to take actions on behalf of users. Both of those goals rely heavily on image understanding. Users expect what's visible on their screen to be accessible to the copilot, and important data is often presented most naturally as an image – think charts, slides, PDFs, etc. In order to take actions, we often need to literally click on buttons or scroll through menus. It would be nice if all these actions were doable via API, but many business-relevant

software has no API or an incomplete API, and controlling software via UIs allows us to keep the user in the loop.

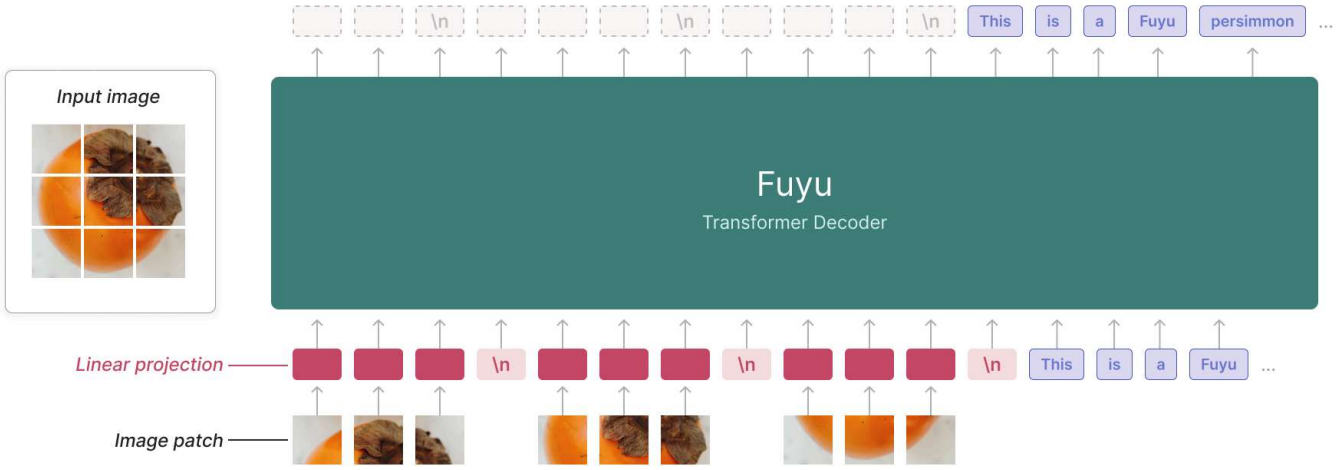


Diagram of the Fuyu model architecture. Fuyu is a vanilla decoder-only transformer with no specialized image encoder. Image patches are linearly projected directly into the first layer of the transformer, bypassing the embedding lookup. This simplified architecture supports arbitrary image resolutions, and dramatically simplifies both training and inference.

Therefore, we need a model that can understand both images and text. Although a lot of progress is being made on this front, nothing is available that suits our precise needs. Existing multimodal models are complicated, both from an architectural perspective and a training perspective. These complications are a liability when it comes to understanding model behavior, scaling models up, and deploying to users.

On the architecture side, other multimodal models involve a separate image encoder, the output of which tends to be connected to an existing LLM via either cross-attention or through some kind of adapter that feeds directly into the LLM's embedding-space. PALM-e, PALI-X, QWEN-VL, LLaVA 1.5, and Flamingo all look more-or-less like this. These models also tend to work on a fixed image resolution. At inference time, all images at greater resolution than this must be downsampled, and all images whose aspect ratio doesn't match must be padded or distorted.

On the training side, other multimodal models tend to have a large number of separate training stages. The image encoder will be trained separately from the LLM on its own tasks, often using a contrastive training objective, which is complicated to implement and reason about. Then, as in e.g. PALI-X, the image encoder and the text decoder (frequently with a bespoke connector network) will be trained together on images at a low resolution for some period of time. At this point, a choice must be made about whether to freeze the weights of each of the components while training. Finally, some models are trained with an extra high-resolution image phase (without which they won't perform well on high-res images).

ADEPT

When scaling up models, it's difficult to reason about how to independently scale each of the above components. Should marginal parameters be allocated to the encoder or the decoder? To which of the training steps should we give the next chunk of compute? We've instead designed a model without these complications.

Architecturally, Fuyu is a vanilla decoder-only transformer with the same details as Persimmon-8B - there is no image encoder. Image patches are instead linearly projected into the first layer of the transformer, bypassing the embedding lookup. We simply treat the normal transformer decoder like an image transformer (albeit with no pooling and causal attention). See the diagram above for more details.

This simplification allows us to support arbitrary image resolutions. To accomplish this, we just treat the sequence of image tokens like the sequence of text tokens. We remove image-specific position embeddings and feed in as many image tokens as necessary in raster-scan order. To tell the model when a line has broken, we simply use a special image-newline character. The model can use its existing position embeddings to reason about different image sizes, and we can use images of arbitrary size at training time, removing the need for separate high and low-resolution training stages.

Together, these changes have dramatically simplified our training and inference experience.

Eval Performance

To sanity-check the architectural changes underlying Fuyu-8B, we chose four of the most commonly-used image-understanding datasets: VQAv2, OKVQA, COCO Captions, and AI2D. VQAv2 and OKVQA are natural image question-answering datasets, COCO is a captioning dataset, and AI2D is a multiple-choice dataset involving scientific diagrams. We compare our models to PALM-e, PALI-X, QWEN-VL, and LLaVA 1.5.

The Numbers

The Fuyu models perform well according to these metrics, even though they are heavily focused on natural images. Fuyu-8B improves over QWEN-VL and PALM-e-12B on 2 out of 3 metrics despite having 2B and 4B fewer parameters, respectively. Fuyu-Medium performs comparably to PALM-E-562B despite having fewer than a tenth as many parameters! PALI-X still performs best on these benchmarks, but it's larger and fine-tuned on a per-task basis. Note that, since these benchmarks are not our main focus, we didn't perform any of the typical optimizations (e.g. non-greedy sampling, fine-tuning for a long time on each dataset specifically, etc).

ADEPT

Eval Task	Fuyu-8B	Fuyu-Medium	LLaVA 1.5 (13.5B)	QWEN-VL (10B)	PALI-X (55B)	PALM-e-12B	PALM-e-562B
VQAv2	74.2	77.4	80	79.5	86.1	76.2	80.0
OKVQA	60.6	63.1	n/a	58.6	66.1	55.5	66.1
COCO Captions	141	138	n/a	n/a	149	135	138
AI2D	64.5	73.7	n/a	62.3	81.2	n/a	n/a

What are these Image-Understanding Benchmarks?

While interacting with these benchmarks we also noticed serious issues. We've developed an in-house eval suite that corresponds more closely to the capabilities we care about, but we thought it was worth elaborating on some of those issues here, given the ubiquity of these benchmarks.

Question Answering Benchmarks

The question-answering datasets are quite flawed - they use a complicated scoring mechanism, require you to respond in a specific format, and are often annotated incorrectly.

Consider the following two images:

OKVQA

Question: "What instrument is the toy bear playing?"

Fuyu's answer: "snare"

OKVQA Score: 0 (all reference answers are simply "drum")

VQAv2

Question: "What type of foods are in the image?"

Fuyu's answer: "fish, carrots"

VQAv2 Score: 0 (reference answers were "hot dogs", "sausages", and "healthy")

For the image on the left from the OKVQA dataset, when asked the question "What instrument is the toy bear playing?", the model responds "snare"—which is clearly true! However, it gets a score of 0, because all of the reference answers are simply "drum". Similarly, for the VQAv2 image on the right, when asked "What type of foods are in the image?", the model corectly responds "fish, carrots", but it also gets a score of 0 because the reference solution list doesn't contain those words.

Captioning Benchmarks

It's also common to evaluate image models using the COCO Captions benchmark. The score used for this benchmark (CIDEr) is based on n-gram similarity to a group of reference captions, which are often poor. We haven't found performance on this benchmark corresponds particularly well to our internal evaluations. In fact Fuyu-Medium is slightly worse by this metric than Fuyu-8B!

For the image below, our model gives the caption "A nighttime view of Big Ben and the Houses of Parliament." This is correct, but it gets a score of 0.4 because it doesn't match any of the reference captions (a good score is over 100).

Fuyu's caption:	"A nighttime view of Big Ben and the Houses of Parliament."
Reference captions:	"A fast moving image of cars on a busy street with a tower clock in the background."
	"Lit up night traffic is zooming by a clock tower."
	"A city building is brightly lit and a lot of vehicles are driving by."
	"A large clock tower and traffic moving near."
	"there is a large tower with a clock on it."
CIDEr Score:	0.4 (No reference caption mentions Big Ben or Parliament)

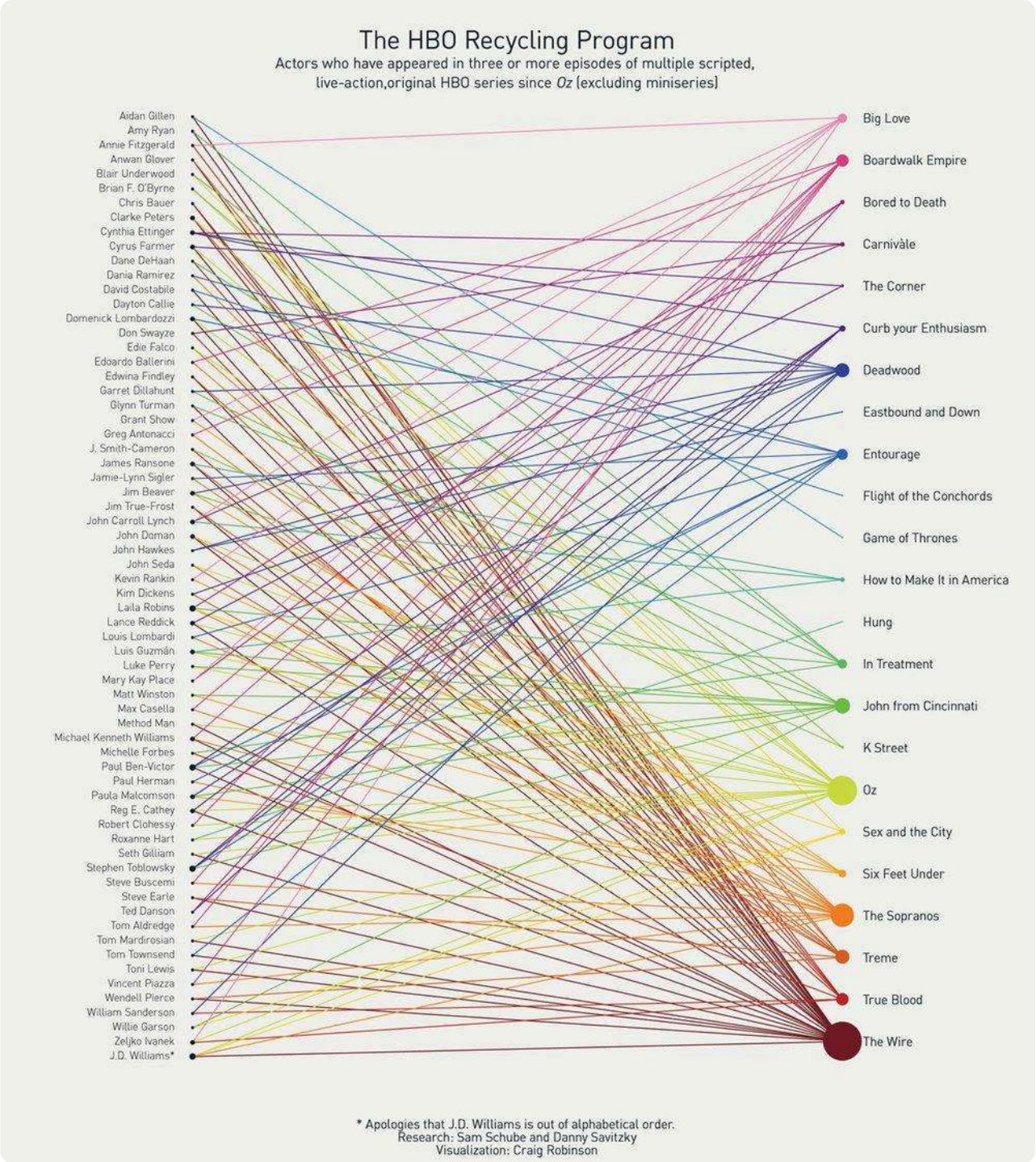
Capabilities

The Fuyu models have several cool capabilities that we preview here, including chart, diagram, and document understanding.

Chart Understanding
ADEPT

Since our product is geared towards assisting knowledge workers, it's important for our model to be able to understand charts and diagrams. Here are some examples.

Fuyu can understand complex visual relationships, such as in the below chart, where it has to trace connections between actors and shows and count them to answer the question.

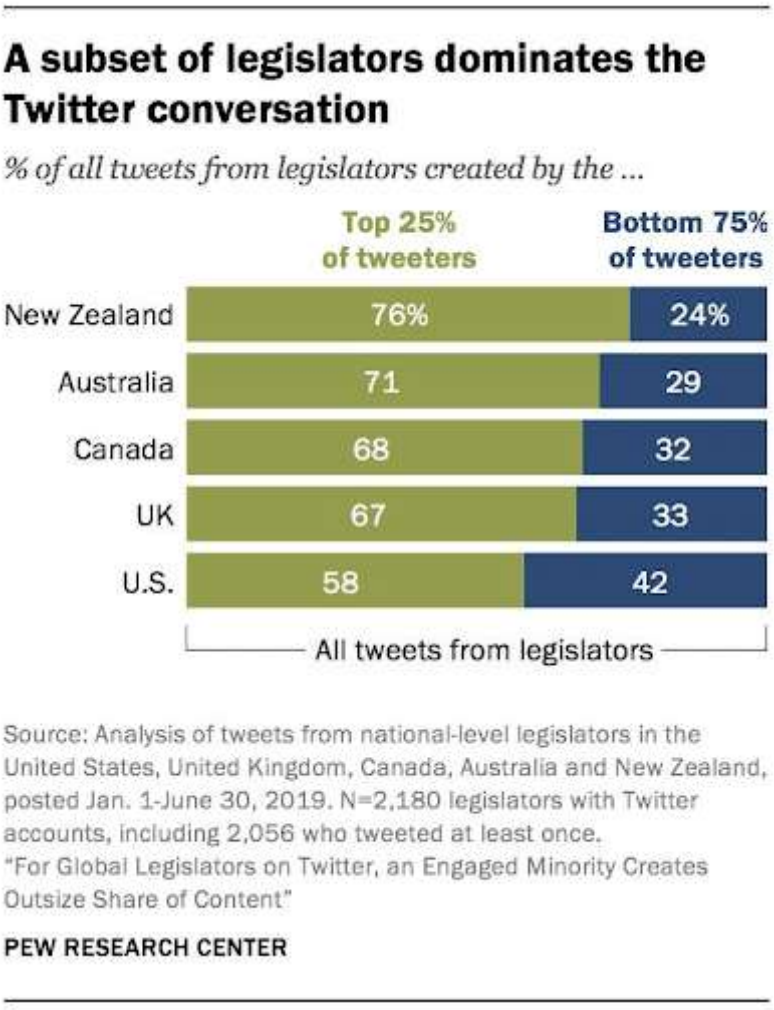


Question: "Aidan Gillen acted in how many series?"

Fuyu's answer: "2"

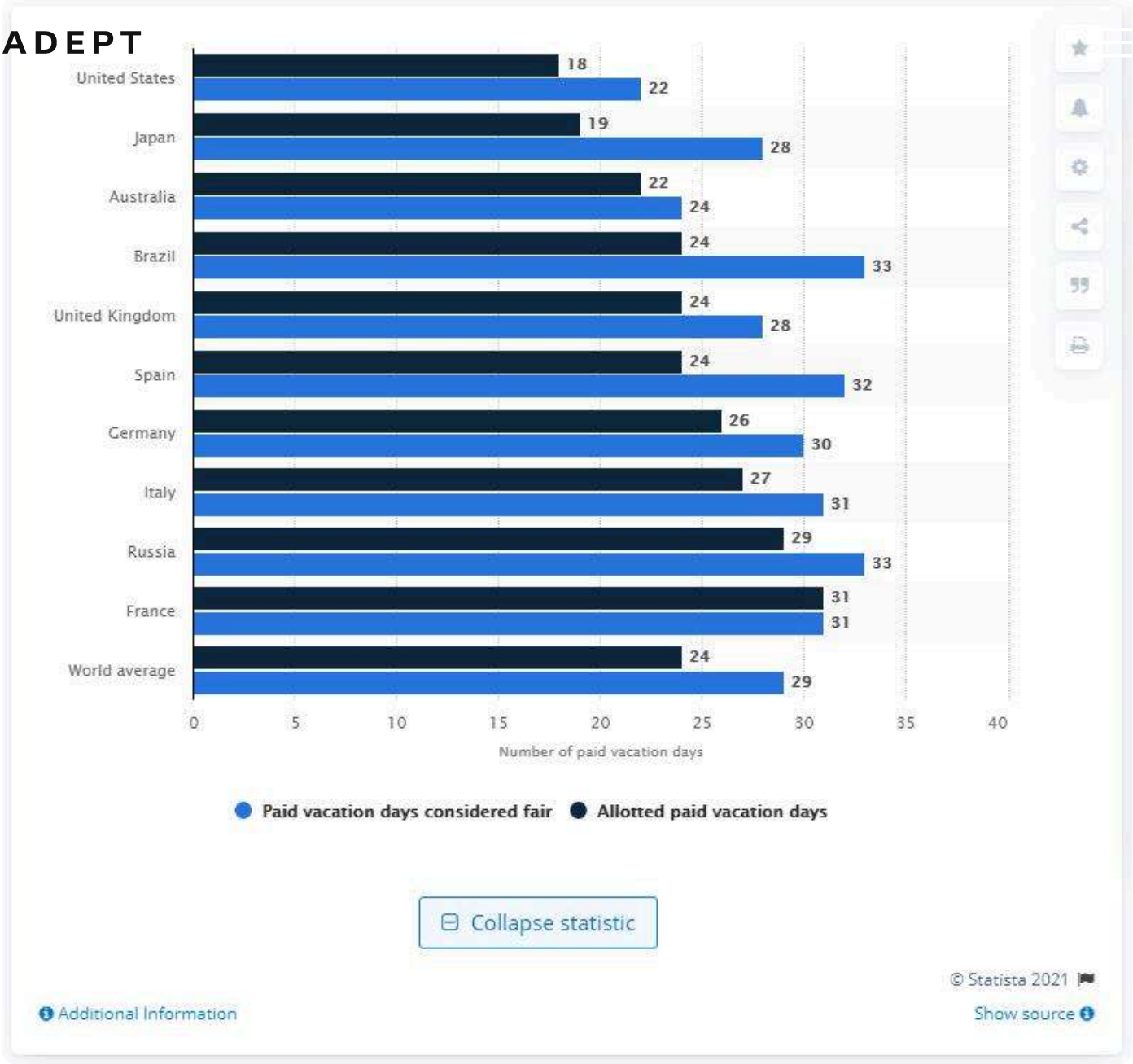
ADEPT

It can also answer nontrivial, multi-hop questions given traditional charts.



Question: "Find missing data of the sequence 24, _ ,32, 33, 42?"

Fuyu's answer: "29"



Question: "What was the fair amount of paid vacation days in the UK?"

Fuyu's answer: "28"

Document Understanding

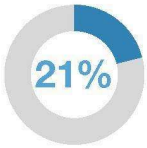
Fuyu can also understand documents — both complex infographics and old PDFs:

ADEPT

WHERE ARE THE JOBS?

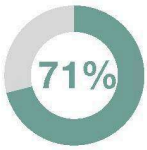
ManpowerGroup Employment Outlook Survey • United States • Q1 2018

11,500
U.S. employers shared their hiring plans for Q1 2018



Plan to hire

Expect to let staff go



Plan to keep workforce levels steady

Are unsure about hiring plans

JOBS OUTLOOK **BY STATE**


Georgia


Hawaii


Florida


Utah


Rhode Island


New Jersey


Washington


Oregon


Alaska

JOBS OUTLOOK **BY METRO AREA**

Cape Coral, Fla.	Los Angeles, Calif.
Ogden, Utah	Phoenix, Ariz.
Chattanooga, Tenn.	Charlotte, N.C.
Providence, R.I.	Albany, N.Y.
Spokane, Wash.	Portland, Ore.
New Haven, Conn.	Cleveland, Ohio

JOBS OUTLOOK **BY INDUSTRY**


Leisure & Hospitality


Transportation & Utilities


Professional & Business Services

Wholesale & Retail Trade

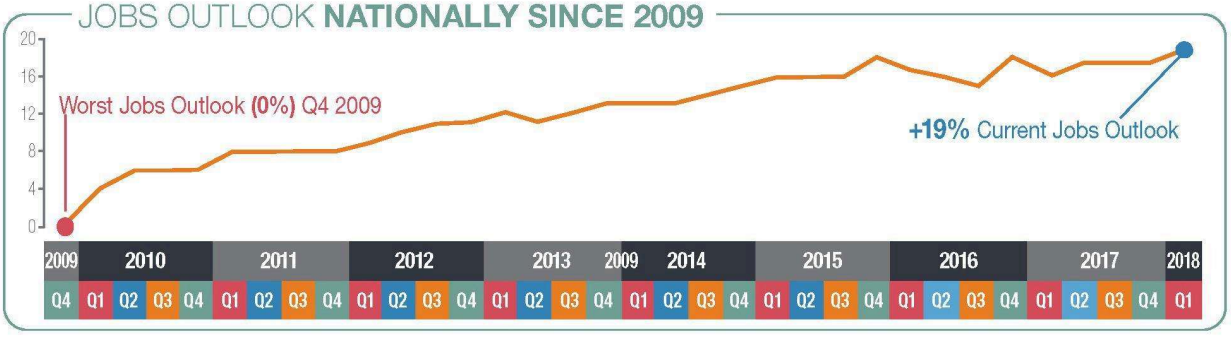

Durable Goods Manufacturing


Government


Other Services


Manufacturing - Nondurable Goods

JOBS OUTLOOK **NATIONALLY SINCE 2009**



Question: "Which is the metro in California that has a good job Outlook?"

Fuyu's answer: "Los Angeles"

ADEPT

Permanent Counter Displays (Continued)

PCD Configurations — Time Savers, New Orleans - 1977

Tested three different PCD configurations with varying capacities. Objective was to determine which generated the highest share for displayed brands.

1. A continuous counter display (original PCD) with add-on units. Capacity was 101 packs.
2. A four sided rotating display (pack spinner) capacity was 118 packs.
3. A smaller display with capacity for 30 packs.

Each display was loaded with various styles of VANTAGE, MORE and/or NOW.

Results

Share of market for the displayed items was 3.6% under each of the configurations tested.

Total cigarette sales were not significantly different for each configuration.

PCD Effectiveness — Bonded Oil, Dayton - 1977

Tested the effect of having PCD's present on total RJR sales. A single PCD with a capacity of 104 packs was used in the test.

Results

Sales of brand styles loaded on the PCD were 5.8% higher than when the PCD was not used.

Incremental sales of display brand styles (purchases by smokers who did not claim the display brands as their usual brand) was 2.7% higher with a PCD present.

Total cigarette category sales increased 0.3% with displays (not significant).

51338 6589

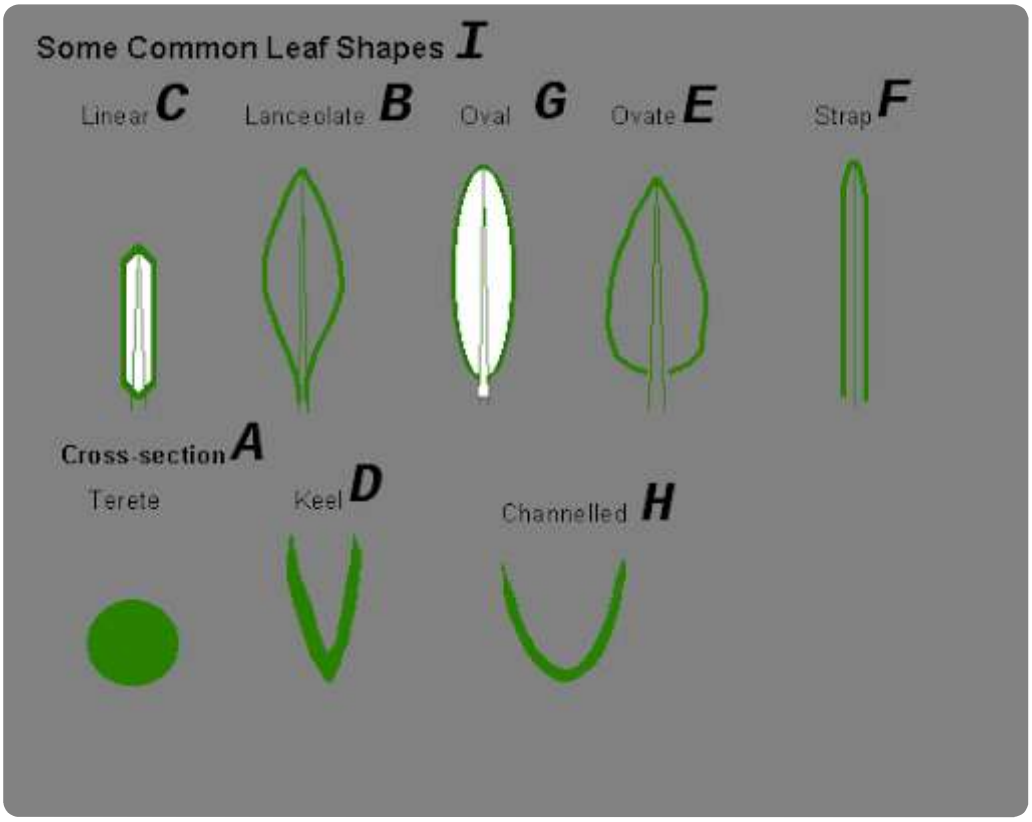
5

Question: "What was the pack spinner capacity?"

Fuyu's answer: "118 packs."

Diagram Understanding
ADEPT

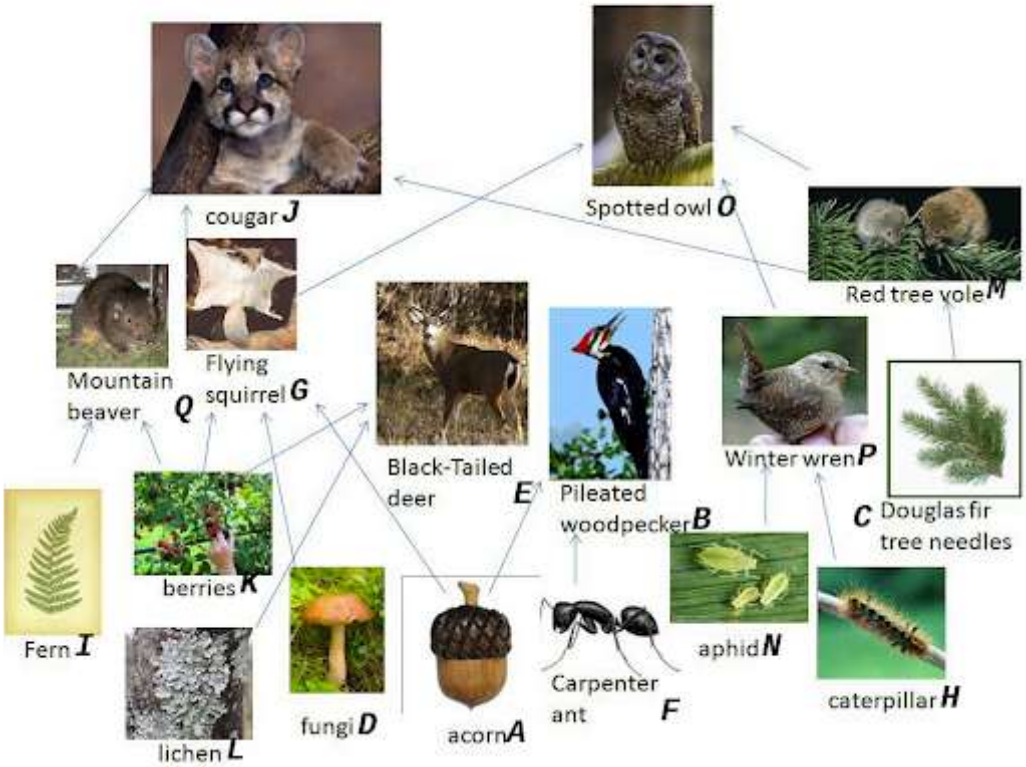
Finally, the model can understand complex relational queries about scientific diagrams:



Question: "What letter does a keel-shaped cross-section look like?"

Fuyu's answer: "The letter V"

ADEPT



Question: "If in the food web shown in the diagram, Douglas fir tree needles are absent, which organism would starve?"

Fuyu's answer: "Red tree vole"

New Capabilities Sneak-Peek

Our internal models (based on Fuyu) have extra capabilities related to our product. In particular,

1. They can reliably perform OCR on high-resolution images
2. They can do fine-grained localization of text and UI elements within those images
3. They can answer questions about images of UIs

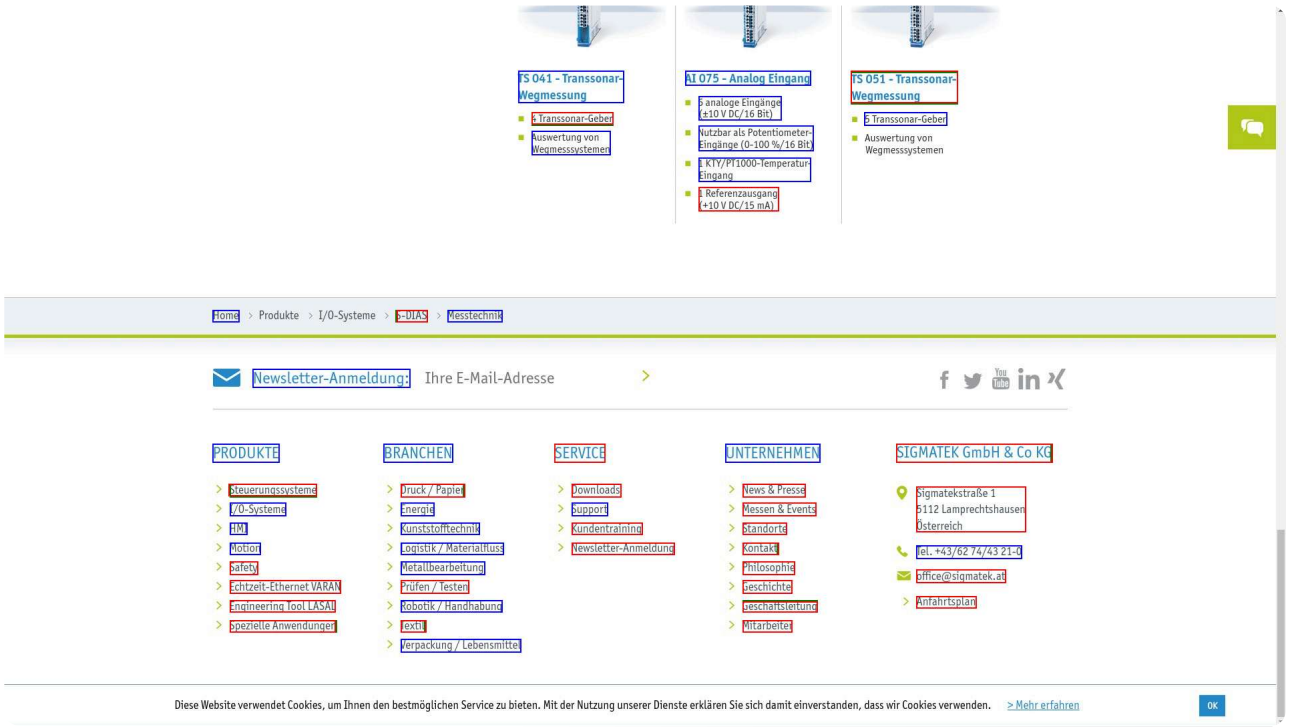
Since these capabilities are built off of the Fuyu model class (and underly our upcoming product release), we thought it would be interesting to preview some of them.

OCR Capabilities

We've trained our internal models to do the following two tasks given an image of a UI:
ADEPT

1. Given a bounding box, tell us what text lies inside that bounding box (`bbox_to_text`)
2. Given some text, return to us the bounding box that contains that text (`text_to_bbox`)

Consider the following 1920x1080 image from one of our validation sets:



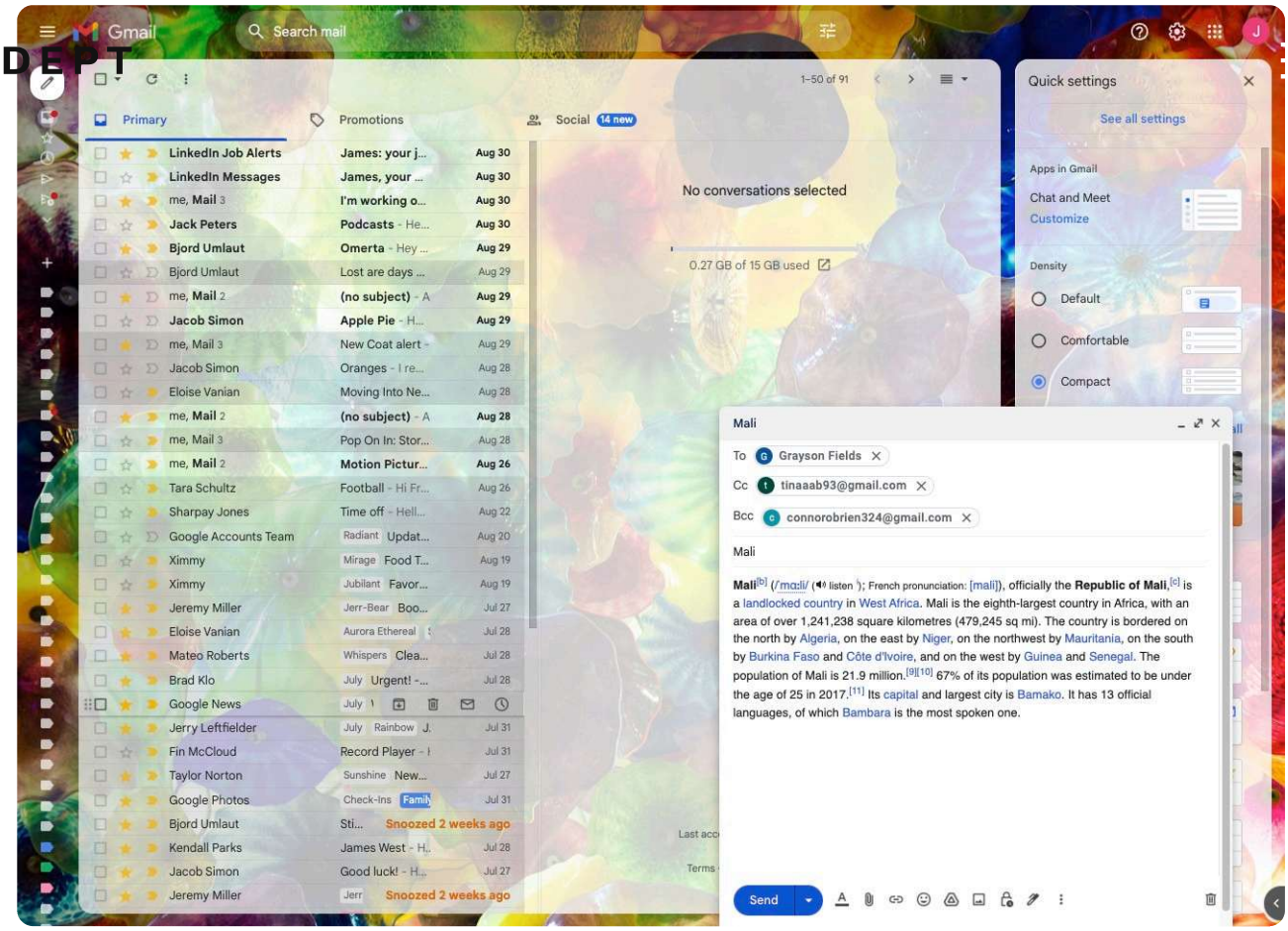
The blue boxes represent bounding box coordinates that have been passed to the model for the `bbox_to_text` task. For this example, the model correctly predicted the text contents of every blue bounding box.

The red boxes represent predicted bounding boxes and green boxes represent target bounding boxes for the `text_to_bbox` task. The model is good enough at bounding box prediction that the red and green boxes overlap almost completely.

Localization and QA Capabilities

The model can also locate things on the screen based on informal text commands, as well as answer detailed factual questions about the contents of UIs:

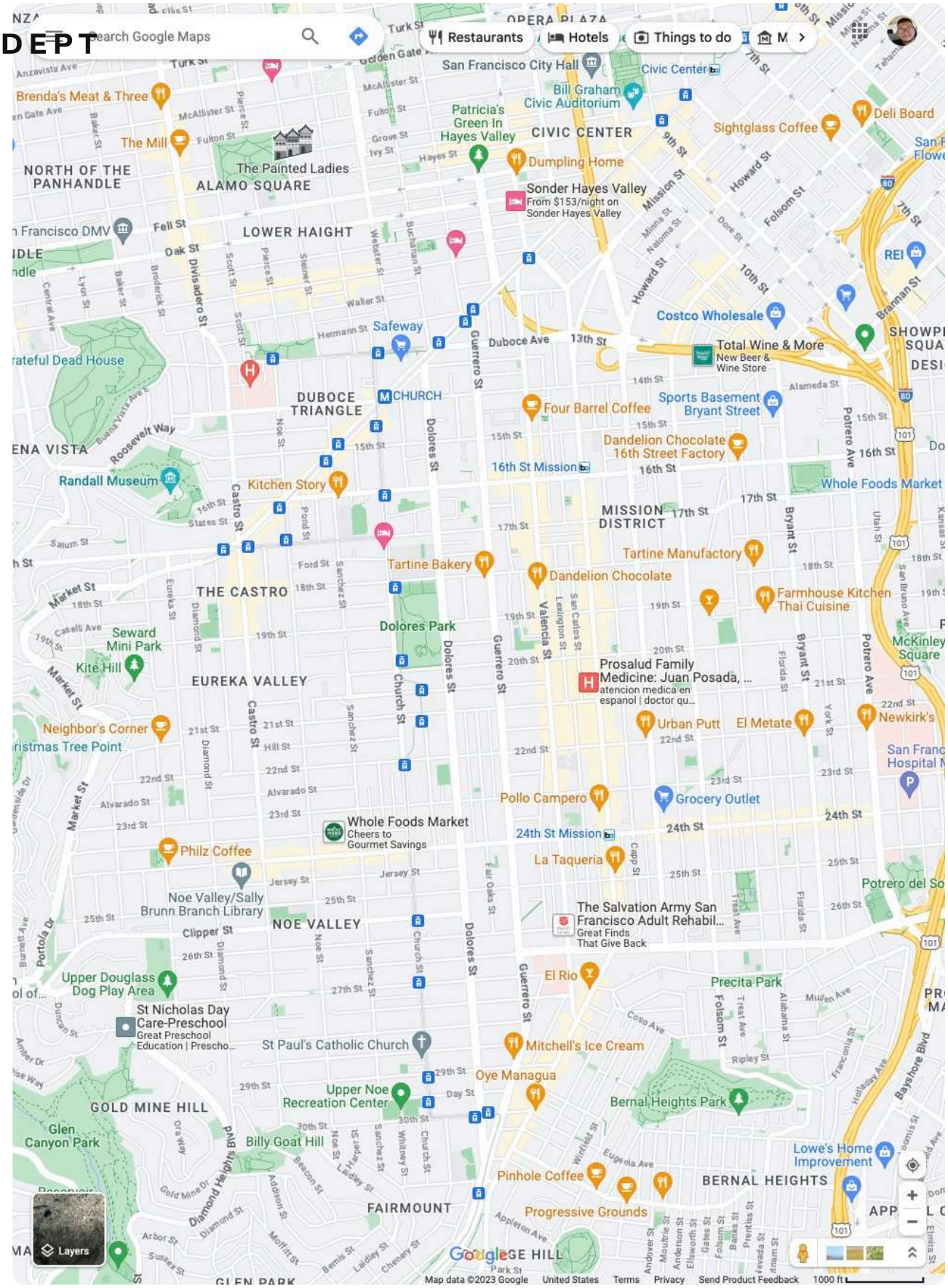
ADEPT



Question: "is the 2nd email starred? ['yes', 'no']"

Fuyu's answer: "no"

Or consider the below example, where the model can interact with Google Maps to correctly answer questions[3].



Question: "is La Taqueria north of the 24th St Mission Bart station?"

Fuyu's answer: "no"

ADEPT

Both the model weights and some example code are on HuggingFace. We look forward to seeing what you build with it, and please reach out if you have any questions. Stay tuned for more on our product alpha, which will incorporate these and other changes and is coming soon!

—

Citation

If you use this model in your work, please use the following BibTeX citation:

```
@misc{fuyu-8b,
  author = {Bavishi, Rohan and Elsen, Erich and Hawthorne, Curtis and Nye, Maxwell and Odena
  title = {Introducing our Multimodal Models},
  url = {https://www.adept.ai/blog/fuyu-8b},
  year = {2023}
}
```

—

Footnotes

1. By "multimodal model," we mean a neural network that can natively see and understand both images and text. ↩

2. Though stay tuned for more on this. ↩

3. Notably, there's no DOM to rely on in this case because the entire map is rendered within a `canvas` tag in the page's HTML. ↩

ADEPT