Assorted topics

Part 2 ["there's MORE!"]

The roundup continues...

Today we'll look at even more 'cutting edge' technologies, and industry implementations - of retrieval, recommendations, knowledge extraction, etc.

This much is for sure - the field of 'IR' is one of the most RAPIDLY changing fields! Why? Because **information** is what runs society :)

External memory to enhance generic chat

The idea of using an LLM as a generic text engine (that has no deep 'domain knowledge) ALONG WITH 'external memory' [a custom 'DB' that contains content knowledge] is rapidly gaining ground!

https://towardsdatascience.com/use-chatgpt-to-query-your-neo4j-database-78680a05ec2 shows how ChatGPT can be used with Neo4j (a graph DB). The idea is this: rather than query Neo4j using its own 'Cypher' query language, we can use natural language instead.

https://tdoehmen.github.io/blog/2023/03/07/quackingduck.html shows to use chat to generate SQL!

RR - Rethinking with Retrieval (https://arxiv.org/pdf/2301.00303) solves reasoning-based tasks by using an external KB (knowledge base).

'Retrieval Transformer' is an approach that also uses external memory to keep the core LLM size DOWN [as low as 4%!!] of a regular LLM - https://jalammar.github.io/illustrated-retrieval-transformer/.

Autonomous task-achieving

Rather than carry on a back-and-forth conversation with an LLM to achieve a task, what if we could specify the task, and let the LLM AUTONOMOUSLY solve it, using sub-goals? This is classic 'agent-based' architecture, an elusive idea in AI thus far!

AutoGPT [eg. https://www.digitaltrends.com/computing/what-is-auto-gpt/] is a brand new approach that does this.

Another radical idea is use LLMs to simulate human behavior, giving rise to 'generative agents': https://arxiv.org/abs/2304.03442.

LLM+tasks+memory -> a 'computer' system!

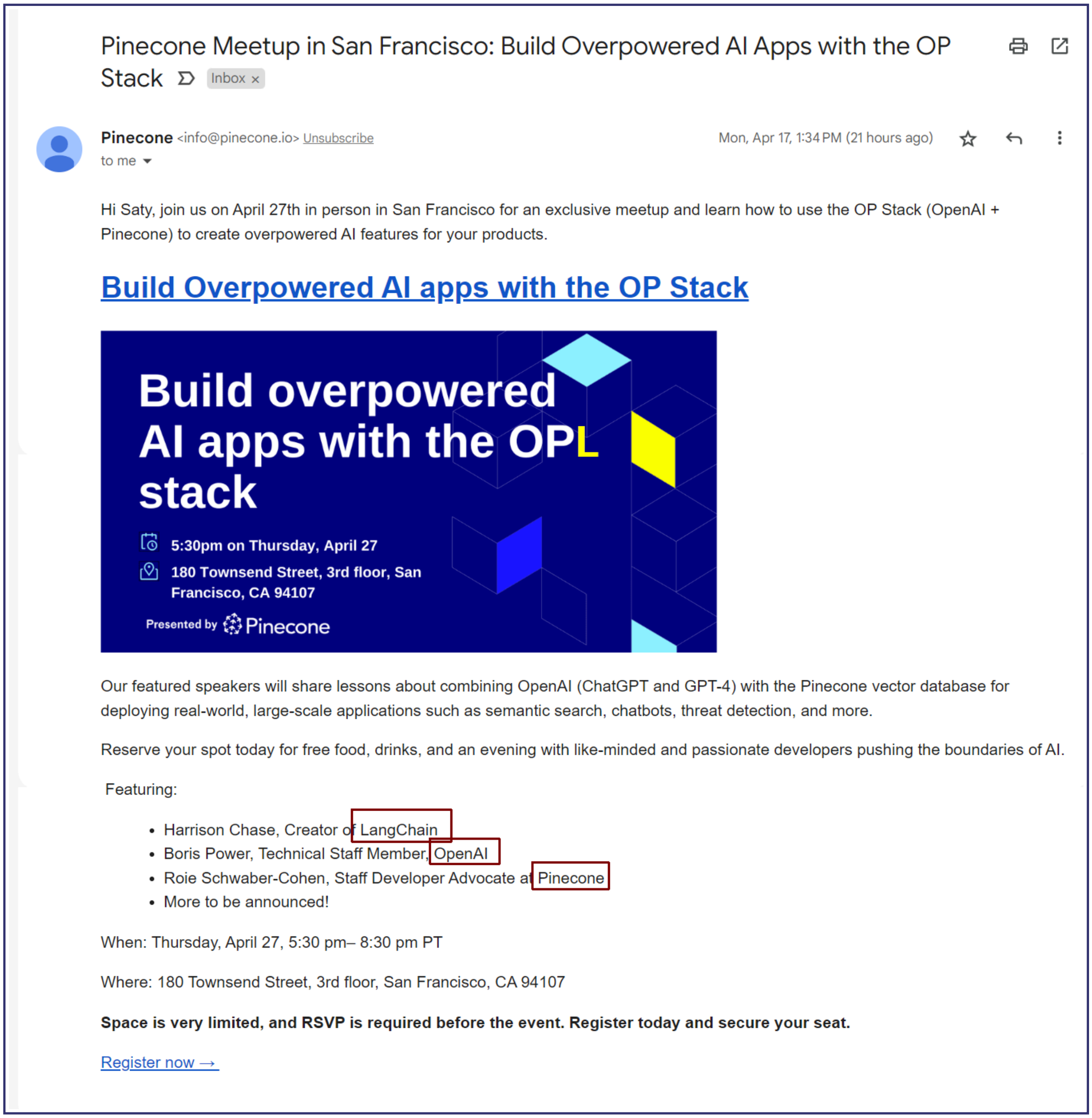
Here is yet another idea: treat the LLM+tasks+DB as a 'computer' (analogous to processor+code+data)!

BabyAGI [eg. https://finance.yahoo.com/news/babyagi-taking-silicon-valley-storm-121500747.html] is such an attempt.

LangChain is a task programming language where we use specific commands in the form of 'templates', to compose our queries, and run( ) them, eg. https://www.pinecone.io/learn/langchain-intro. SudoLang is another such task-spec language.

'OPL' is the name we could give to such a stack comprised of O(penAI)+P(inecone)+L(angChain), eg. https://towardsdatascience.com/building-llms-powered-apps-with-opl-stack-c1d31b17110f

Also :)



Build overpowered AI apps with the OPL stack
5:30pm on Thursday, April 27
180 Townsend Street, 3rd floor, San Francisco, CA 94107
Presented by Pinecone
Inbox ×

Pinecone Meetup in San Francisco: Build Overpowered AI Apps with the OP Stack

Pinecone <info@pinecone.io> Unsubscribe
to me

Mon, Apr 17, 1:34 PM (21 hours ago)

Hi Saty, join us on April 27th in person in San Francisco for an exclusive meetup and learn how to use the OP Stack (OpenAI + Pinecone) to create overpowered AI features for your products.

Build Overpowered AI apps with the OP Stack

Our featured speakers will share lessons about combining OpenAI (ChatGPT and GPT-4) with the Pinecone vector database for deploying real-world, large-scale applications such as semantic search, chatbots, threat detection, and more.

Reserve your spot today for free food, drinks, and an evening with like-minded and passionate developers pushing the boundaries of AI.

Featuring:

- Harrison Chase, Creator of LangChain
- Boris Power, Technical Staff Member, OpenAI
- Roie Schwaber-Cohen, Staff Developer Advocate at Pinecone
- More to be announced!

When: Thursday, April 27, 5:30 pm– 8:30 pm PT

Where: 180 Townsend Street, 3rd floor, San Francisco, CA 94107

Space is very limited, and RSVP is required before the event. Register today and secure your seat.

Register now →

NER

'NER' (Named Entity Recognition) as you know, is a useful NLP information-related task - given text, or images, or video, or audio, what person/place/thing/... can we identify?

We can use BERT for NER, eg. via PyTorch.

Topic modeling

BERTopic is a topic-modeling technique based on BERT.

Here is a guide.

KG construction

Knowledge graphs (KGs) are an excellent form of knowledge representation, since they are well structured (eg via (s,p,o) triplets). https://medium.com/@dallemang/llms-closing-the-kg-gap-29feee9fa52c shows how we can use ChatGPT to create KGs from plain (ie unstructured text).

Recommendation engines

REs are univerally useful, across multiple domains.

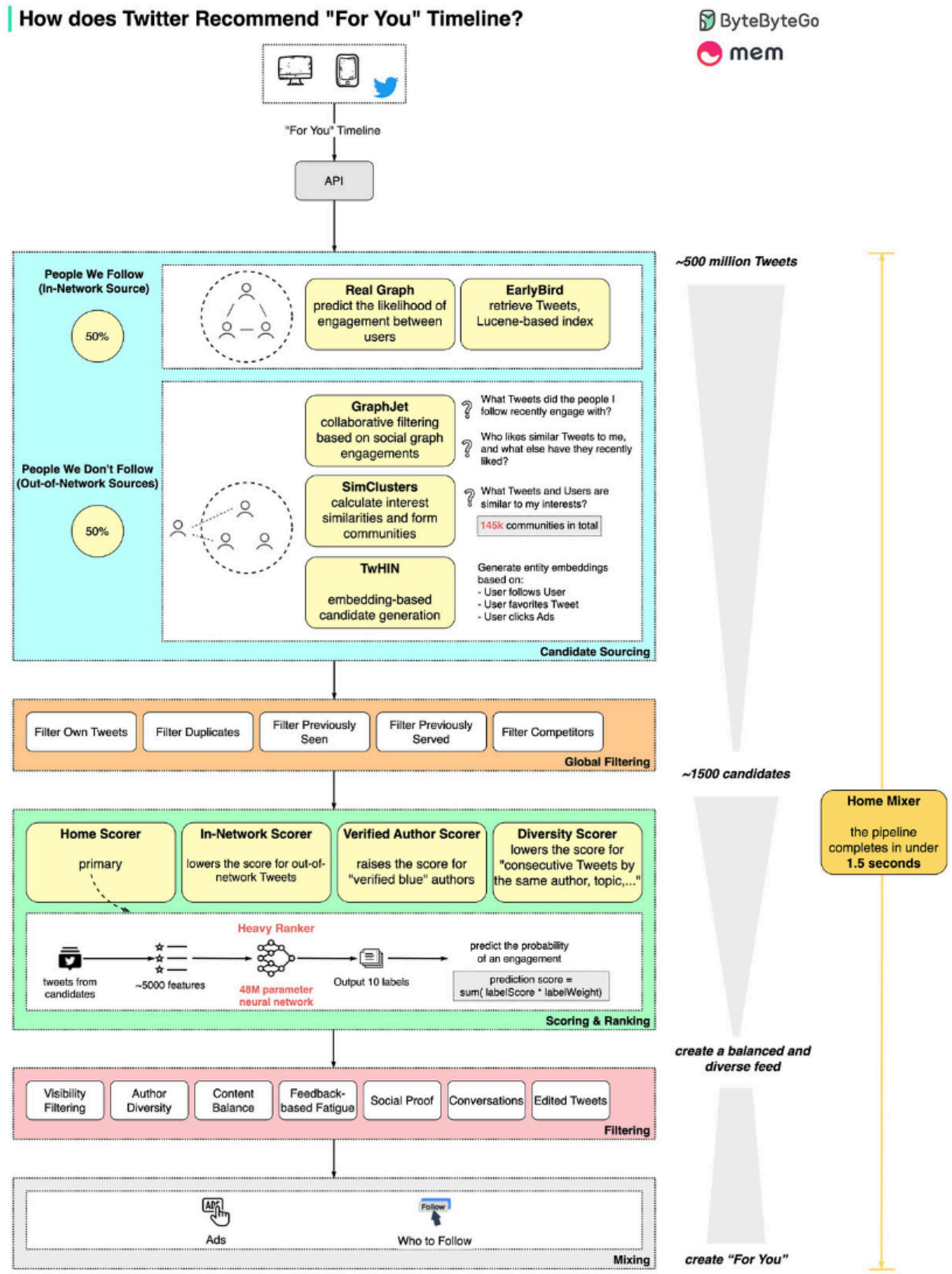
Monolith is TikTok's RE: https://analyticsindiamag.com/tiktok-parent-bytedance-reveals-its-sota-recommendation-engine/.

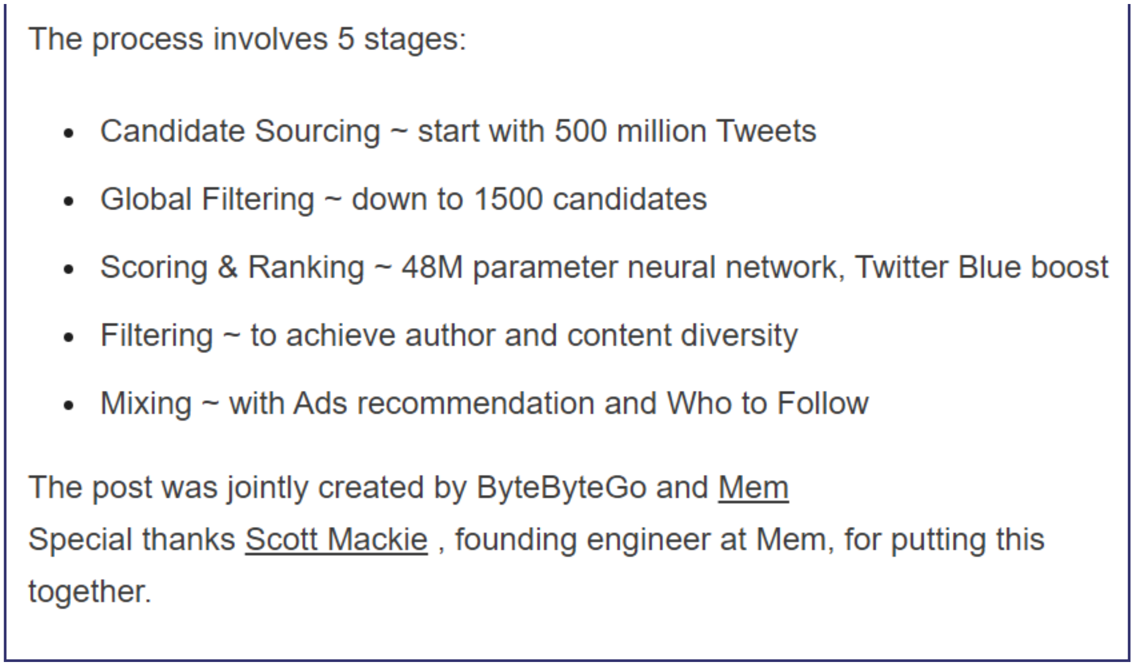
Twitter's RE:

How does Twitter recommend “For You” Timeline in 1.5 seconds?

We spent a few days analyzing it.

The diagram below shows the detailed pipeline based on the open-sourced algorithm.


How does Twitter Recommend "For You" Timeline?
ByteByteGo
mem
"For You" Timeline
API
People We Follow (In-Network Source)
50%
Real Graph predict the likelihood of engagement between users
EarlyBird retrieve Tweets, Lucene-based index
People We Don't Follow (Out-of-Network Sources)
50%
GraphJet collaborative filtering based on social graph engagements
What Tweets did the people I follow recently engage with?
Who likes similar Tweets to me, and what else have they recently liked?
SimClusters calculate interest similarities and form communities
What Tweets and Users are similar to my interests?
145k communities in total
TwHIN embedding-based candidate generation
Generate entity embeddings based on:
- User follows User
- User favorites Tweet
- User clicks Ads
Candidate Sourcing
Filter Own Tweets
Filter Duplicates
Filter Previously Seen
Filter Previously Served
Filter Competitors
Global Filtering
Home Scorer primary
In-Network Scorer lowers the score for out-of-network Tweets
Verified Author Scorer raises the score for "verified blue" authors
Diversity Scorer lowers the score for "consecutive Tweets by the same author, topic,..."
Heavy Ranker
tweets from candidates
~5000 features
48M parameter neural network
Output 10 labels
predict the probability of an engagement
prediction score = sum(labelScore * labelWeight)
Scoring & Ranking
Visibility Filtering
Author Diversity
Content Balance
Feedback-based Fatigue
Social Proof
Conversations
Edited Tweets
Filtering
Ads
Who to Follow
Follow
Mixing
~500 million Tweets
~1500 candidates
create a balanced and diverse feed
create “For You”
Home Mixer
the pipeline completes in under 1.5 seconds



The process involves 5 stages:

- Candidate Sourcing ~ start with 500 million Tweets
- Global Filtering ~ down to 1500 candidates
- Scoring & Ranking ~ 48M parameter neural network, Twitter Blue boost
- Filtering ~ to achieve author and content diversity
- Mixing ~ with Ads recommendation and Who to Follow

The post was jointly created by ByteByteGo and Mem
Special thanks Scott Mackie , founding engineer at Mem, for putting this together.

Lyft's RE: https://eng.lyft.com/the-recommendation-system-at-lyft-67bc9dcc1793.

Clustering

We looked at K-means clustering. Here are others you can look up:

- t-SNE, eg. https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a
- HDBSCAN, eg. https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html
- UMAP, eg. https://umap-learn.readthedocs.io/en/latest/clustering.html

Search

Here is an assortment of 'search' related items:

- this search looks for CC (Creative Commons) licensed content (images, audio)
- 'manifold search' uses manifold learning for similarity searches, eg. https://scikit-learn.org/stable/modules/manifold.html
- NN searches over a graph (rather than metric space): https://research.yandex.com/blog/graph-based-nearest-neighbor-search
- Redis-based AI-driven (vector) search: https://partee.io/notes/2022-9-13-SDSC-talk/
- Discord search (using inverted indexing): https://sukhadanand.medium.com/how-discord-indexes-billions-of-messages-f242e605e47c

NeRF, AR LBS

Standard LBS retrieves addresses, maps.

A new Google Maps update will retrieve immersive views, made possible by fusing together numerous distinct photos and aerial views, and seamlessly rendering them using NeRF [more here].