

Go Hybrid: Find the Results **LIKE** Can't

PyOhio 2026 · Jamal Hansen

jamalhansen.com

| **Before we start**

The content of this talk is my own and does not represent the views of my employer.

About me

San Antonio, Texas

Data scientist

I work in SQL and Python

You will see both today, and both will be approachable

Meet Butterscotch



| Question

How many of you work with data?

| Survey



| Data

- Documents
- Source code
- Chat messages (Slack, Teams, Discord)
- Email

| Text

Answers exist in unstructured data

| Challenges of finding meaning in text

Find text that means you walked a dog

Search for `walk` :

- `walk`ed Butterscotch in park
- quick morning stroll with the dog
- took him jogging
- long run at the park

Search for `run` :

- We are `run`ning low on dog treats
- Butterscotch is `wrun`g out after a long day

Words are meaningful and messy

Wag & Wash



The best neighborhood dog spa for eight years.

Team keeps great records in the system.

They write a note about each visit.

Butterscotch loves going to this spa.

60-minute walk completed.

Bruno was such a good boy this afternoon; we took a nice stroll through the park

More Examples

Not all records are as easy to read.

```
DW-60. Bruno good. No issues.
```

```
GRP-WLK done. pup did well.
```

| The search for premium dogs

Purr Palace will expand their services from cats to dogs.

They open across the street from [Wag & Wash](#) in two weeks.

Wag & Wash wants to identify premium clients with a loyalty offer.

| What is "premium"?

Wag & Wash sells two kinds of visits.

- **Regular**: walks, day care, a basic bath.
- **Premium**: the spa treatment. Full grooming packages, dental cleanings, deshedding.

Premium clients pay for extras.

Those are the clients Purr Palace wants to steal.

| The same service, multiple ways to say it

The only record of premium dogs is in the visit notes

Examples of premium grooming visits:

- GRM-60, BATH, TRIM, NAILS. Done.
- Our groomer did the full works today
- Tooth scaling and polish completed
- Gave the pup a wash and a tidy-up

Lots of ways to describe premium visits.

448 records without any structure

Reading everything does not scale

You already know LIKE

Python



```
"groom" in note
```

SQL

```
WHERE notes LIKE '%groom%'
```

LIKE is **Ctrl+F** for your data.

It finds the exact characters in text.

This is the tool I reached for first.

Find records like `groom`

Searching for `%groom%` finds:

- "booked her usual grooming"
- "groomed by our senior stylist"

Also finds the wrong records:

- "Great grooming weather today!! Took Daisy on her usual walk"

And never finds:

- "gave the pup a wash and a tidy-up"
- "the full treatment"
- GRM-60

Possibly misses:

- "GROOM"

LIKE misses real records and matches wrong ones.

So you add logic.

```
WHERE notes LIKE '%groom%'  
      OR notes LIKE '%bath%'  
      OR notes LIKE '%spa%'  
      OR notes LIKE '%GRM%'  
AND notes NOT LIKE '%weather%'
```

Or probably this...

```
WHERE ((notes LIKE '%groom%' OR notes LIKE '%grm%')
      AND notes NOT LIKE '%weather%')
      OR (notes LIKE '%bath%' OR notes LIKE '%bath%')
      OR ((notes LIKE '%spa%' AND notes LIKE '%spa%')
          AND NOT (notes NOT LIKE '%GRM%'))
      OR notes LIKE '%tidy%' OR notes LIKE '%tidy-up%' OR notes LIKE '%tidyup%'
      OR notes LIKE '%GRM-60%' OR notes LIKE '%grm60%'
AND (1=1)
-- weather fix, DO NOT REMOVE (jake, 2019)
AND notes LIKE '%weather%'
```

To add all the logic, you have to read all the records.

This is what we're trying to avoid.

LIKE matches characters but misses meaning

Search by meaning to find the missing records

Vector search presents a solution to this problem.

It reads a note and produces an embedding: numbers that capture what the note means.

So `walked`, `stroll`, and `jogging` all group together.

...and it understands that `running` low on treats has a different meaning

```
from sentence_transformers import SentenceTransformer

model = SentenceTransformer("all-MiniLM-L6-v2")
vec = model.encode("our groomer did the full works")
```

Vector search solves problems with LIKE

It understands natural speech so our query becomes:

- professional grooming and spa services for dogs

It can even handle some typos:

- **shampooo**, conditioning, **finishin** spray, coat looks **sooo** fluffy n **beutiful**
- bath blow dry **trimmmed** ear clean and nails.

This looks like the answer.

Vectors also fail



A problem note:

"GRM-60, BATH, TRIM, NAILS, EAR CLN. Done."

- GRM-60 has no meaning to embed
- Terse notes ("Nails trimmed") lack context

Vectors understand meaning and misses concepts it's not trained on

Why not both? Hybrid?

	Codes, IDs, exact terms	Paraphrase, typos, synonyms
LIKE	Match	Miss
Vector	Miss	Match

So, the plan is to run both and merge the results.

```
results = like_ids | vector_ids    # union, done?
```

So let's use LIKE and vectors. What could go wrong?

| Unioning results doesn't work

LIKE returns results that matched.

Vectors give a score from 0 to 1.

The results are not comparable.

I took a wrong turn somewhere



So I stopped guessing and measured

448 rows describing visits. Scored against an answer key.

| LIKE was the wrong way to match keywords

LIKE asks: *is this string present?*

Best Matching 25 (BM25) asks: *how much is this note actually about these terms?*

The math dates to 1994. Today it is the ranking engine inside Lucene and Elasticsearch.

```
PRAGMA create_fts_index('service_logs', 'id', 'notes');
```

The real keyword search is BM25

BM25 does things LIKE cannot

It allows multiple keywords.

- grooming bath spa nails teeth trim shampoo overnight boarding

It ranks.

- The best matches come back first.

It weights.

- Rare words count more: deshedding > dog.

It stems.

- groom, grooming, and groomed have the same root word and are grouped.

Does BM25 find Butterscotch (premium dogs)?



"GRM-60, BATH, TRIM, NAILS, EAR CLN. Done."

Tool	Verdict	Why
LIKE <code>%groom%</code>	MISS	the word "groom" is not there
Vector	MISS	a string of codes has little meaning
BM25	HIT	matches <code>BATH</code> , <code>TRIM</code> , <code>NAILS</code>

BM25 finds the other premium services too



"Butterscotch completed an overnight boarding stay with us and was an excellent guest."

Tool	Verdict	Why
LIKE <code>%groom%</code>	MISS	the word "groom" is not there
Vector	MISS	Meaning of <code>boarding</code> doesn't match to <code>grooming and spa</code>
BM25	HIT	<code>boarding</code> is one of my query terms, ranks #4

One query with every service word spans every premium service. LIKE needs a separate pattern per service.

BM25 still misses



"our groomer did the full works today, ears and coat all sorted"

Tool	Verdict	Why
LIKE %groom%	HIT	groom is in groomer
BM25	MISS	groomer does not stem to groom
Vector	HIT	understands the meaning

The stemmer folds groom / grooming / groomed but not groomer .

Vectors can help to catch what it misses.

| Hybrid, done right

The hybrid result that works is BM25 + vector.

Fuse ranks.

Because both searches return results in order, each list votes by position.

Top results on both lists are strong.



Bringing the data together with Hybrid search

Each search hands in a ranked ballot. Records high on either ballot rise.
Records on both rise fastest.

Reciprocal Rank Fusion

```
def rrf(bm25_ids, vec_ids, k=60):  
    scores = {}  
    for rank, id in enumerate(bm25_ids):  
        scores[id] = scores.get(id, 0) + 1 / (k + rank + 1)  
    for rank, id in enumerate(vec_ids):  
        scores[id] = scores.get(id, 0) + 1 / (k + rank + 1)  
    return sorted(scores, key=scores.get, reverse=True)
```

`k=60` is a standard default.

BM25 and vector results combine based on rankings.



Place your bets

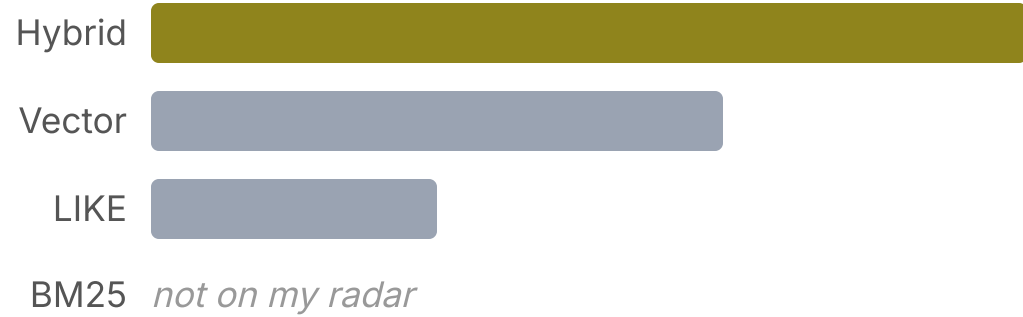
Four tools. 24 premium clients. Which one found the most?

QUERIES

Using sample queries that match the search

- LIKE `groom`
- Vector `professional grooming and spa services for dogs`
- BM25 `grooming bath spa nails teeth trim shampoo overnight boarding`

What I thought the results would be



My understanding was that $\text{LIKE} + \text{Vector} = \text{Hybrid}$
Each step would improve results.

What the data actually said

My bet · before measuring

BM25 *not on my radar*

Hybrid

Vector

LIKE

The data ·

BM25

Hybrid

Vector

LIKE

Tool	Dogs returned	Correct	Wrong
BM25	16	15	1
Hybrid	16	14	2
Vector	20	15	5
LIKE	15	12	3

If BM25 is better, why use Vector?

- A cleaner top of the list so the records hybrid ranks highest are almost all correct.
- Extended word-form coverage
 - groomer , dental , undercoat : the places where stemming quietly breaks.

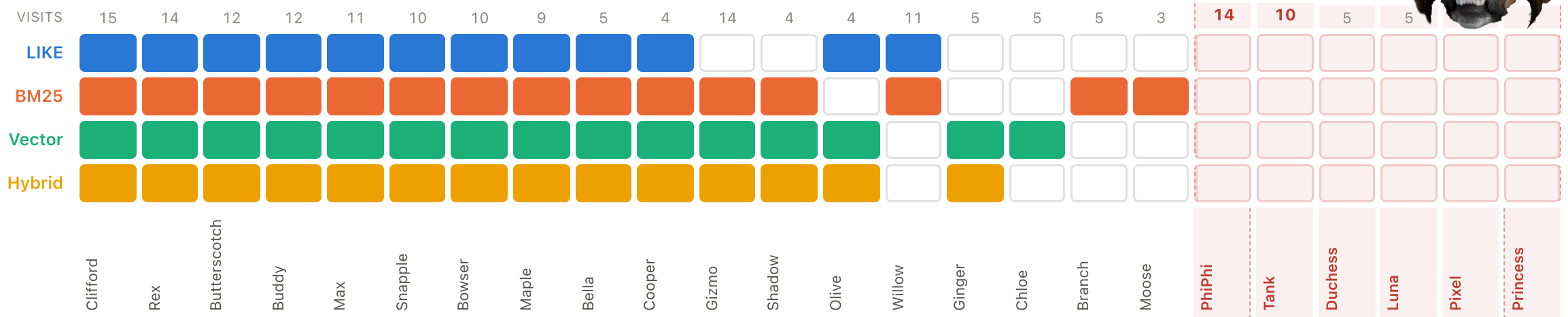
Each tool finds its own set of records.

Four tools, four jobs



Tool	Reach for it when
LIKE	you know the exact string
BM25	codes, shorthand, exact terms (the workhorse)
Vector	paraphrase, typos, word-form variation
Hybrid	you want the cleanest top of the list

Who each tool actually found



Insights

- 10 premium dogs are found by every tool
- 8 where the tools split
- 6 aren't found by any method
 - two of these have many visits

A closer look at some disagreements

Clients where the tools disagree

Client	Found by	Comments
Olive	all but BM25	BM25's best record ranks #46.
Ginger	vector, hybrid	Wrong words: de-shed , undercoat , argan oil
Branch	BM25 only	Terse shorthand so vectors have nothing to embed (vector rank #200+)
Moose	BM25 only	Dental visit and vectors don't read teeth-cleaning as "spa" (vector rank #200+)

Six premium dogs that weren't found

Each dog's best record, and the tool that comes closest to surfacing it:

Client	#	Best record	Closest rank
PhiPhi	14	"full spa package: shampoo, conditioning..."	#41 · BM25
Luna	5	"did lu's nails today, trimmed real nice"	#43 · BM25
Pixel	4	"SPA. Shampew, conditioning, LVS."	#47 · Vector
Tank	10	"full spa day: shampooo, conditioning"	#68 · Vector
Duchess	5	"...got her nails trimmed up nicely"	#82 · BM25
Princess	3	"...brush out and trim, coat gorgeous"	#133 · Vector

PhiPhi came in **14 times** and her best record lands at **#41** — one slot under the top-40 line. Butterscotch is found because a **single** record spikes to #3.

| What to know about search methods

- Words are meaningful and messy
- LIKE matches characters but misses meaning
- Vectors understand meaning and misses concepts it's not trained on
- The real keyword search is BM25
- BM25 and vector results combine based on rankings.
- Each tool finds its own set of records.



| What I learned on my data journey

- Stop chaining LIKE-based logic to find data.
- BM25 is worth trying on my next project
- Don't make assumptions; measure
- Answers are not always clear-cut

| Try it on a note of yours

Give me any note about a dog.

We will see which tools find it in the database.



Thanks

Jamal Hansen

jamalhansen.com

Notebook + code: github.com/jamalhansen/go-hybrid

