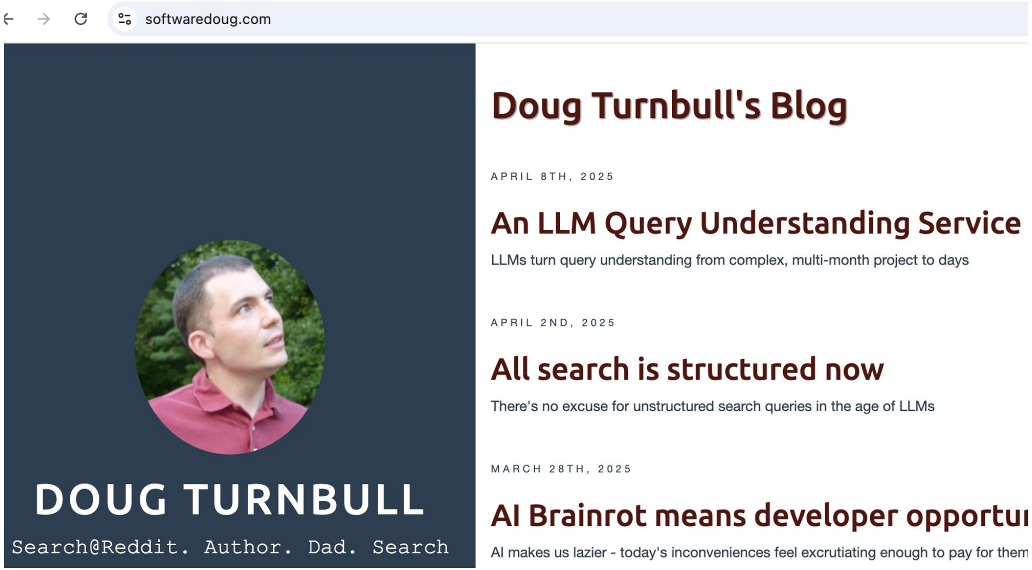
LLM-driven Relevance Engineering

Haystack 2025

Obligatory Bio Slide

👋 Hi I’m Doug
(@softwaredoug everywhere)

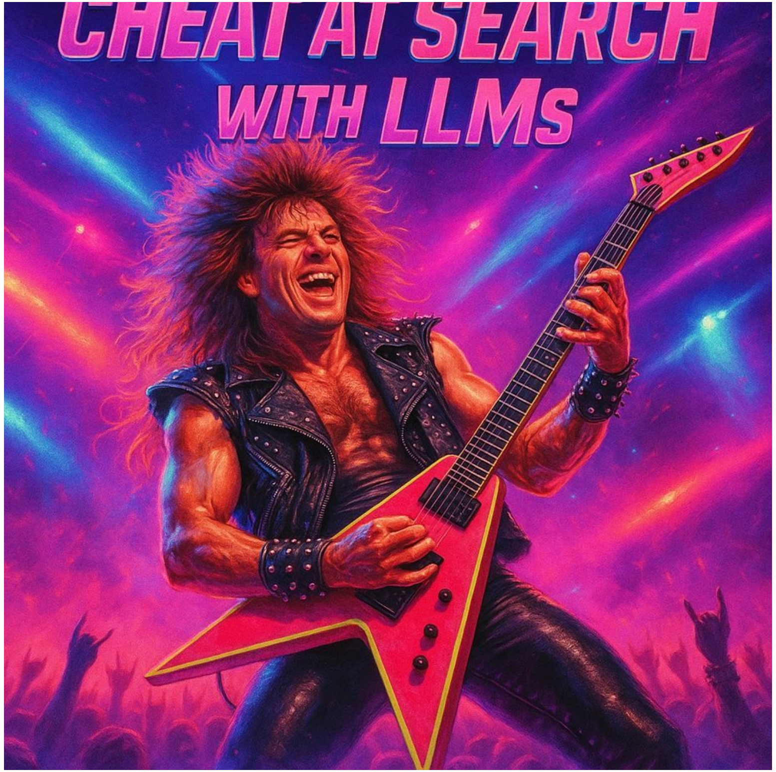
I blog here: http://softwaredoug.com



~~Live in Concert~~

Training course:

LLM Query understanding, content, and judging



Obligatory Plug

https://maven.com/softwaredoug/cheat-at-search

Discount Code: **searchybird** good through Apr

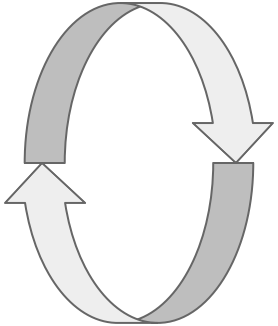
(EXPIRES 1.5 weeks)

Disclaimers, etc

- Going to talk about my specific path - what I did at Daydream (but on open dataset)
- IE not the end-all be-all of this topic *by far!*

Search labeling very time consuming

Labelers



Labelers

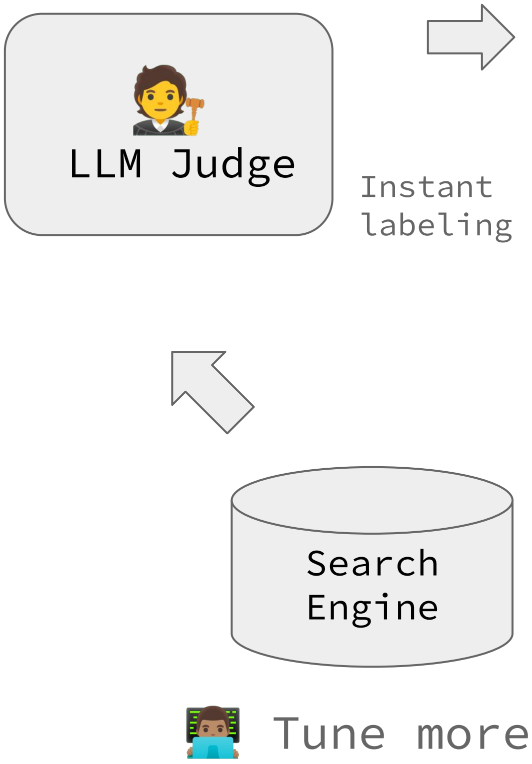
Labelers

Query	Document	Rating (1-5)
red shoes		5
red shoes		5
red shoes		3
red shoes		3
red shoes		1

New algorithm, results haven’t been labeled

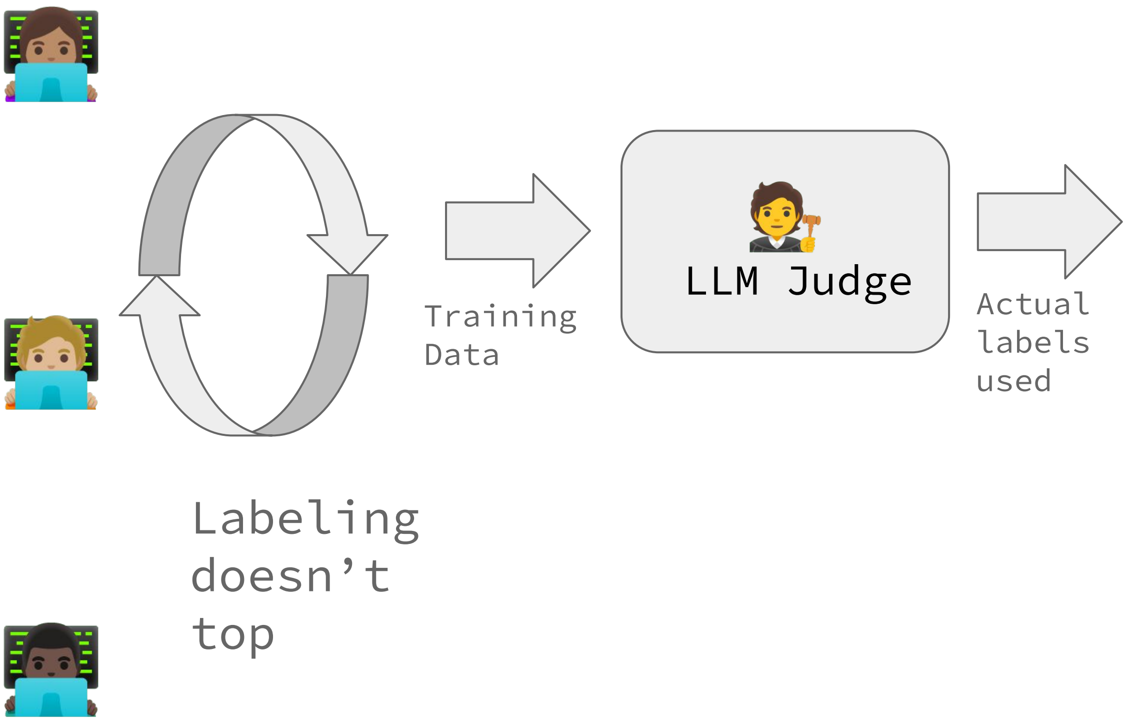
Query	Document	Rating (1-5)
red shoes		5
red shoes		??
red shoes	🩰	??
red shoes	🛼	??
red shoes	👖	1
red shoes	👠	5

LLM Judge: Tighter feedback loop



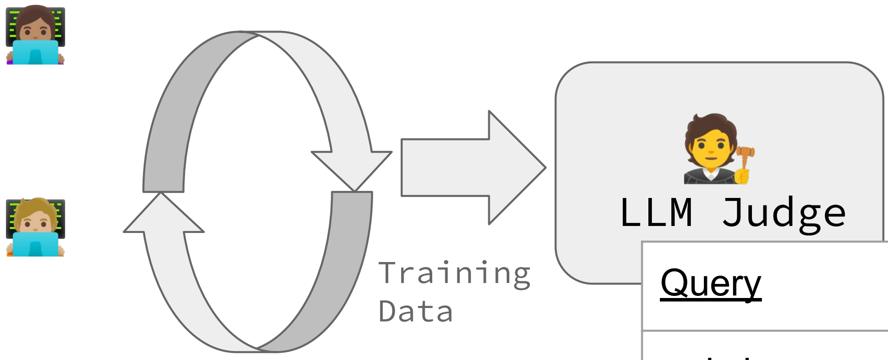
Query	Document	Rating (1-5)
red shoes		5
red shoes		4
red shoes	🩰	4
red shoes	🛼	2
red shoes	👖	1
red shoes	👠	5

Human labeling guides LLM judge



Query	Document	Rating
red shoes		5
red shoes		4
red shoes		4
red shoes		2
red shoes		1
red shoes		5

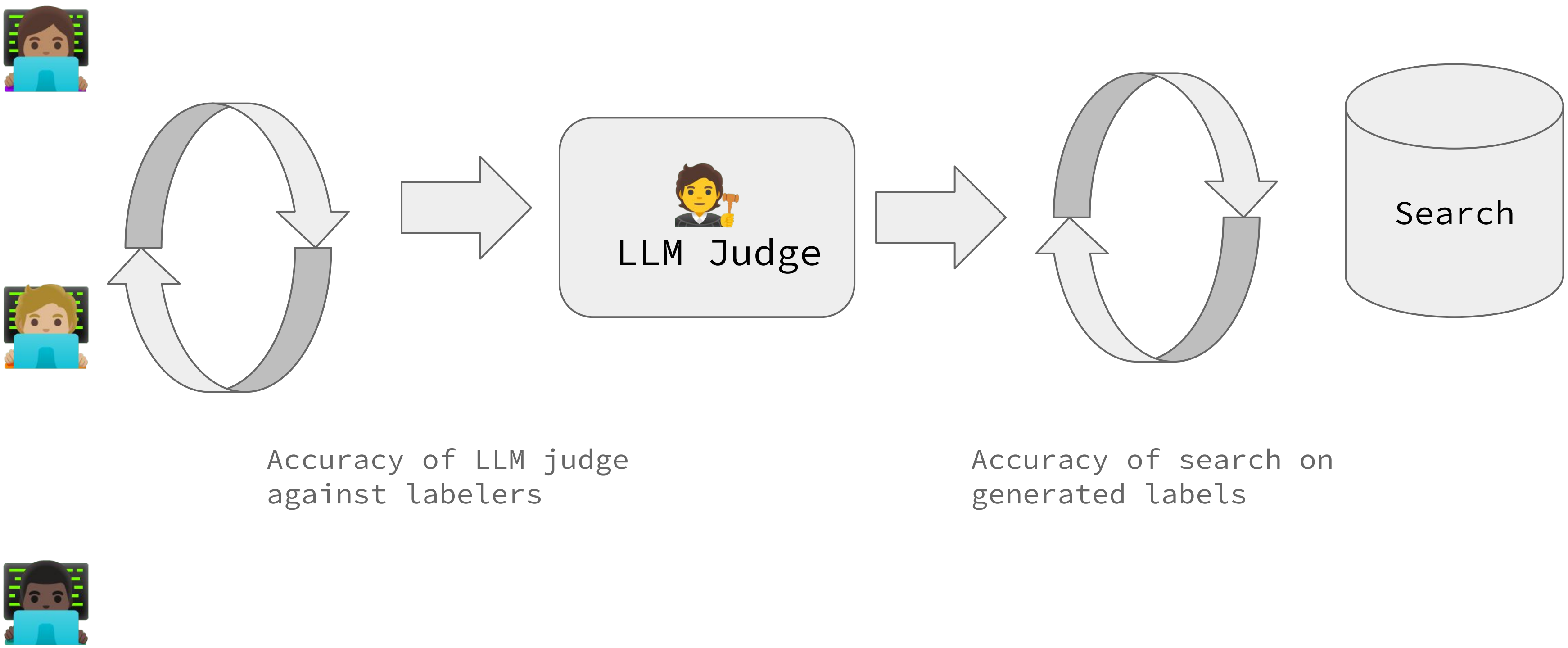
Does our judge generalize?



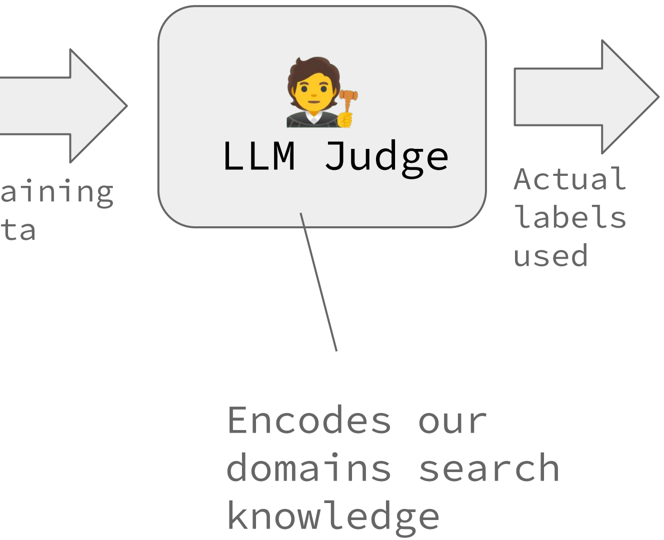
Is our judge any good?

Query	Document	Human Label	LLM Label	Delta
red shoes	👟	5	4	1
red shoes	👟	4	4	0
red shoes	🩰	4	3	1
red shoes	🛼	2	2	0
red shoes	👖	1	3	2
red shoes	👠	5	4	1

Now we have two feedback loops

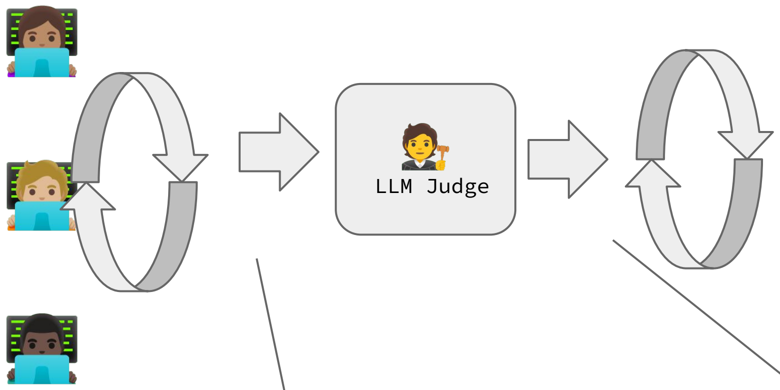


Human labeling program -> LLM training program



Query	Document	Rating (1-5)
red shoes		5
red shoes		4
red shoes		4
red shoes		2
red shoes		1
red shoes		5

This is an _*organizational process*_ NOT magic



Understanding limits of LLM judge as labeling source

Organizational discipline to manage raters
(though fewer needed)

Classic judgment list problems:

- Categorical distinctions add noise (a 1 vs 2)
- (A lot of tuning required to reduce that noise)

True of humans + LLMs!

(ESPECIALLY smaller models)

```
You are a search relevance
judge. Evaluate the relevance
of the following product based
on the following criteria

5 - exact match, user would buy
4 - almost exact match, user
might buy…
...
```

Actual goals:

- Guidance during manual tuning of search
- Minimize cost of human labels (by doing 1-5 labels)
- Handle noisiness of human categorical labels

Switching to pairwise LLM judge

vs

Which of these two products are more relevant for the search query "red adidas sambas"?

Product LHS Name: adidas sambas
Product LHS Description: These red adidas sambas are a fashionable fitness shoe
Product RHS Name: pink adidas sambas
Product RHS Description: Check out these hot pink shoes, you'll love them!

Agent: LHS

Pros

- Pairwise choices less noisy [1]

Downsides:

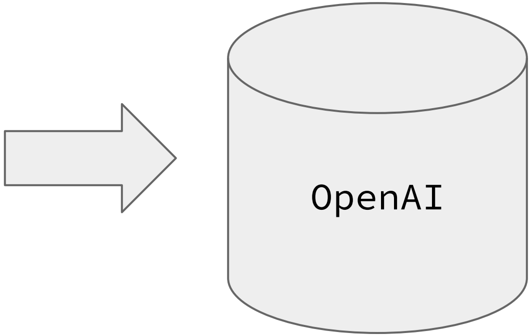
- Instead of rating N, you rate NxN

1 – “When is it better to Compare Than To Score”
https://arxiv.org/pdf/1406.6618

Annoyingly NxN calls…

Isn’t this a lot of OpenAI calls?

vs



Enter Apple Silicon

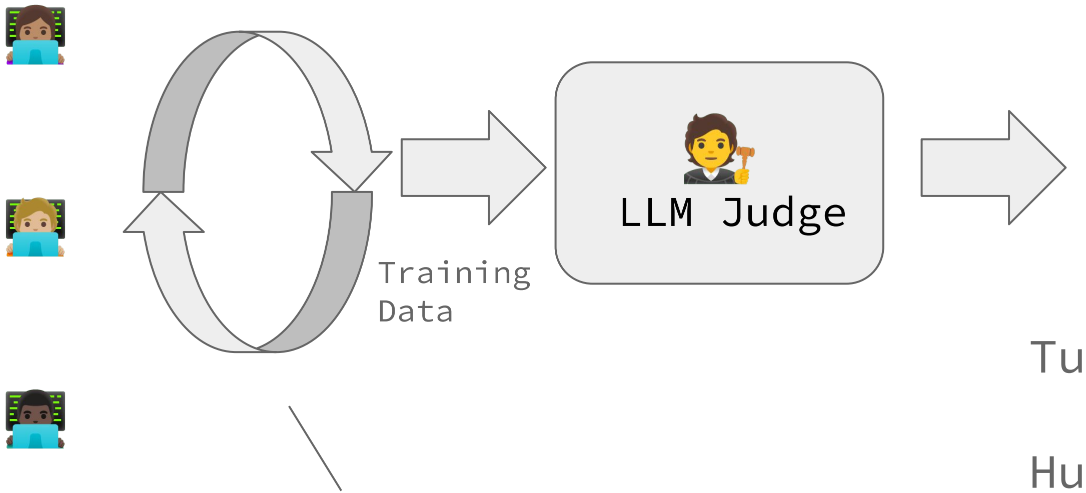
VS 

RHS!

Dearest laptop: which is most relevant?

https://github.com/softwaredoug/local-llm-judge/

Optimizing our Judge



LHS or RHS?

Turns into MANY comparisons:

HumanRating(LHS) >
HumanRating(RHS)?

Still 1-5 labels
to speed up rating

Pairwise: An Alternative Flow

Query	Document
red shoes	
red shoes	
red shoes	
red shoes	
red shoes	

VS

Query	Document
red shoes	
red shoes	
red shoes	
red shoes	
red shoes	

Optimizing == What prompt to use?

(What to add?)

Which of these two products are more relevant for the search query "red adidas sambas"?

Product LHS Name: adidas sambas
Product LHS Description: These red adidas sambas are a fashionable fitness shoe
Product RHS Name: pink adidas sambas
Product RHS Description: Check out these hot pink shoes, you'll love them!

- Product Image?
- Category?
- Reviews
- ??

More accurate decision?

(or just confusion)

Let's try some prompts out…

(Using - **W**ayfair **AN**otated **D**ata**S**et)

Attribute	Precision (Compared to human raters. N=1000)
Just Name	75.08%
Just Description	70.31%
Just Category	74.60%
Just Class	70.50%

https://github.com/softwaredoug/local-llm-judge/

Strategies

(Using - **W**ayfair **AN**otated **D**ata**S**et)

Attribute	Precision (Compared to human raters. N=1000)
Just Name	75.08%
Just Description	70.31%
Just Category	74.60%
Just Class	70.50%
All Fields	78.10%

https://github.com/softwaredoug/local-llm-judge/

Checking twice…

Swap LHS/RHS and check for agreement

Attribute	Precision (Compared to human raters. N=1000)	Precision / Recall
Just Name	75.08% / 100%	87.99% / 58%
Just Description	70.31% / 100%	76.58% / 72.60%
Just Category	74.60% / 100%	86.1% / 69.7%
Just Class	70.50% / 100%	87.76 / 58.0%
All Fields	78.10% / 100%	91.72% / 65.2%

We can take it a step further

Observe what our judges are doing:

Query/Pair	Title Judge	Desc Judge	Human Pref
Red shoes  vs 	LHS	Neither	LHS
Red shoes  VS 	RHS	LHS	RHS
…	…	…	…

Train model on each judge to predict human…

Query/Pair	Title Judge	Desc Judge	Human Pref
Red shoes  vs 	LHS	Neither	LHS
Red shoes  **VS** 	RHS	LHS	RHS
…	…	…	…

`Pref = f(Title_Judge, Desc_Judge, ...)`

Features

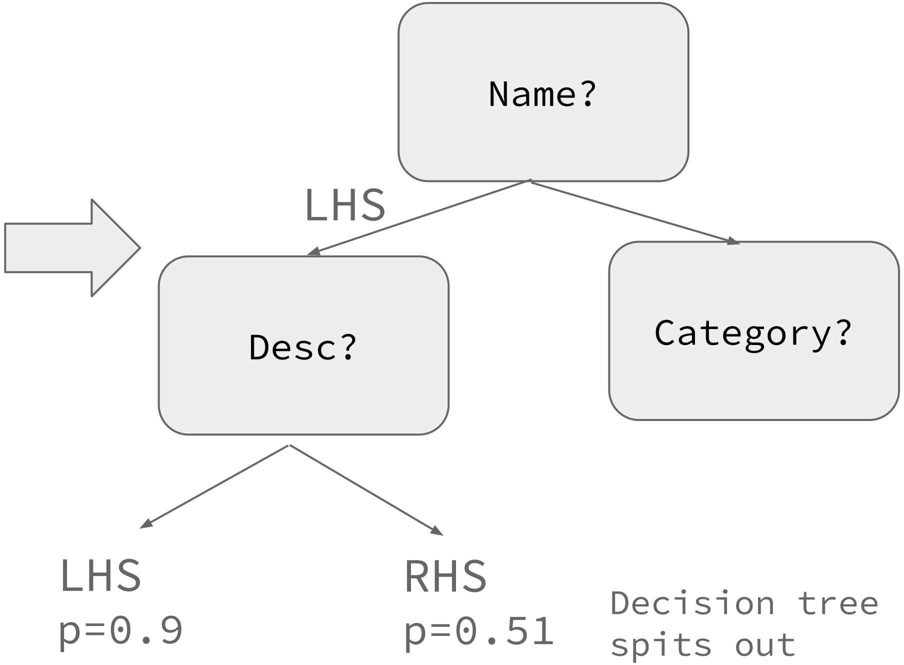
Predictor

Which of these judges has biggest impact?

Query/Pair	Title Judge	Desc Judge	Human Pref
Red shoes vs	LHS	Neither	LHS
Red shoes VS	RHS	LHS	RHS
…	…	…	…

Features

(Decision tree model)



Decision tree spits out probability of LHS/RHS

Which of these judges has biggest impact?

Query/Pair	Title Judge	Desc Judge	Human Pref
Red shoes  vs 	LHS	Neither	LHS
Red shoes  VS 	RHS	LHS	RHS
…	…	…	…

Features
(the judges)

```
clf = DecisionTreeClassifier()
clf.fit(train[feature_columns],
        train['human_preference']
```

Which of these judges has biggest impact?

Query/Pair	Title Judge	Desc Judge	Human Pref
Red shoes  vs 	LHS	Neither	LHS
Red shoes  VS 	RHS	LHS	RHS
…	…	…	…

Trying to predict

```
clf = DecisionTreeClassifier()
clf.fit(train[feature_columns],
        train['human_preference']
```

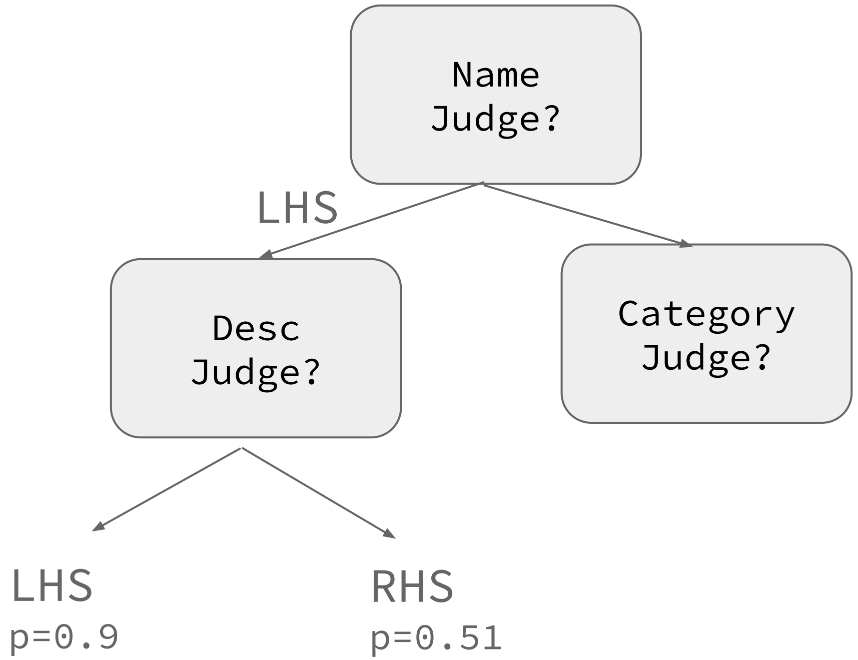
Adding decision tree model

Attribute	Checking Once (Compared to human raters. N=1000)	Checking Twice Precision / Recall
Just Name	75.08% / 100%	87.99% / 58%
Just Description	70.31% / 100%	76.58% / 72.60%
Just Category	74.60% / 100%	86.1% / 69.7%
Just Class	70.50% / 100%	87.76 / 58.0%
All Fields	78.10% / 100%	91.72% / 65.2%
Decision Tree		94.10% / 44.9%

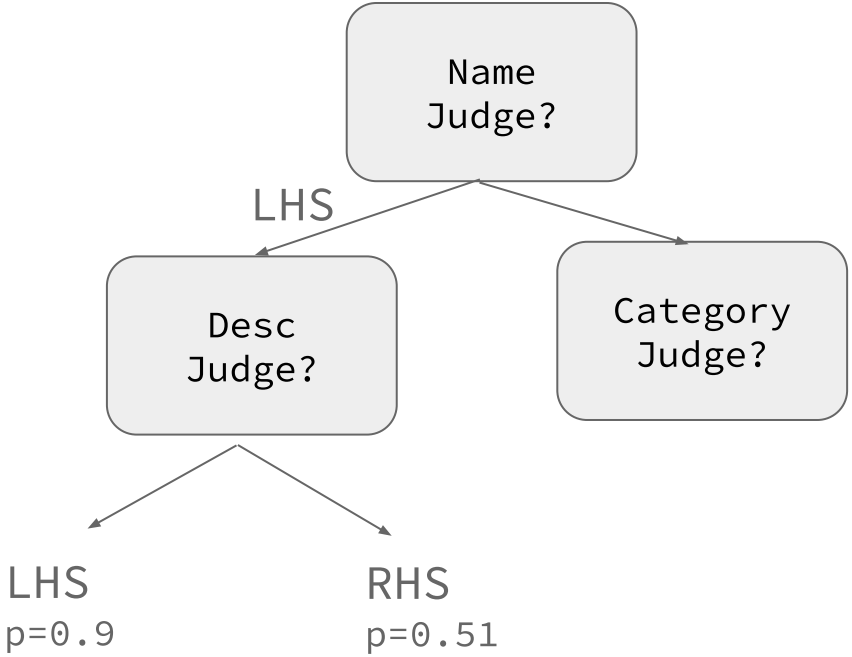
Trained on above, p > 0.9 from decision tree

Accuracy isn’t the point

Feature Insight…



Feature Insight…



These are the original features

A lot of relevance work:

Current index:

```
{
    “Name”: “Adidas Sambas”
    “Description”: “These
cool red shoes…”
    “Category”:
Footwear/blah/blah
    “Class”: “Athletic
Footwear”

}
```

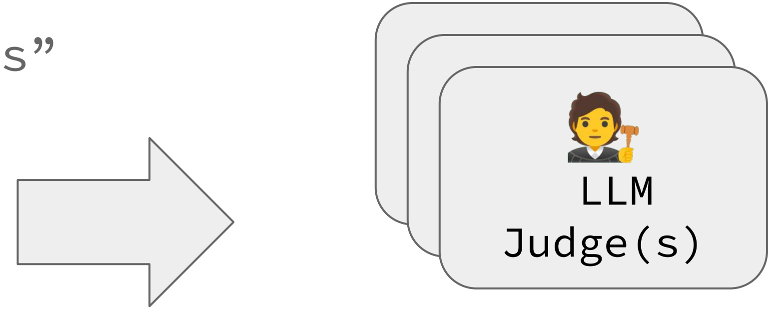
...10% of our queries mentioned color...

LLM Judge(s) have blind spot with color

```
Current index:

{
    “Name”: “Adidas Sambas”
    “Description”: “These
cool red shoes…”
    “Category”:
Footwear/blah/blah
    “Class”: “Athletic
Footwear”

}
```



Eval for
Red adidas sambas

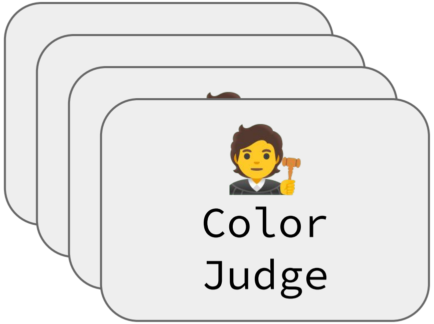
No color to eval!

But if we added color to the index…

```
Current index:

{
    “Name”: “Adidas Sambas”
    “Description”: “These
cool red shoes…”,
    “Category”:
Footwear/blah/blah,
    “Class”: “Athletic
Footwear”,
    “Color”: “Red”

}
```



Eval for
Red adidas sambas

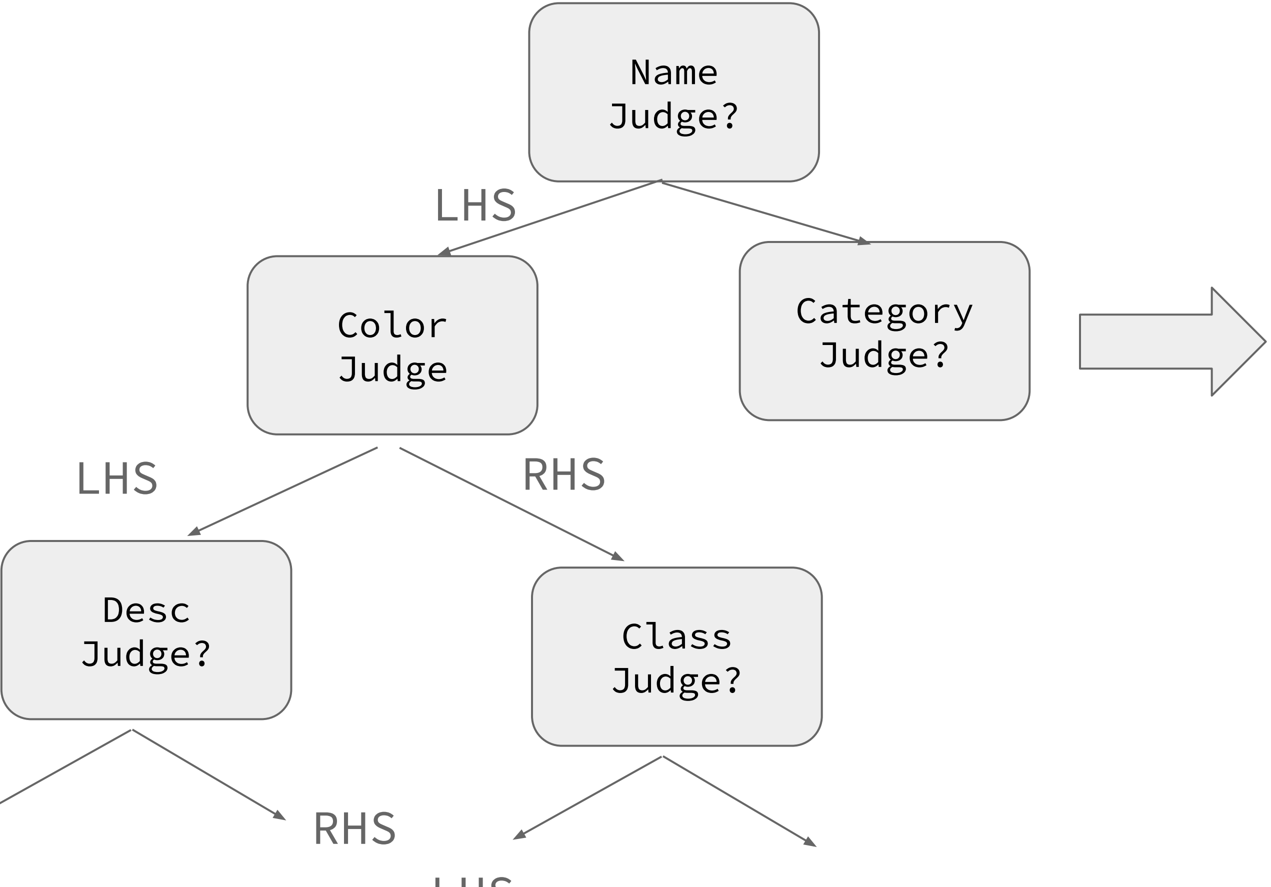
With a dumb judge just to do...

Color
Judge

```
Which of these is more
relevant for query: red
adidas sambas

LHS Color: red
RHS Color: brown
```

We could see its impact...



This model:

- Better predicts human labels by 2%
- Show color has outsized impact on decision tree eval

Adding decision tree model

Attribute	Checking Once (Compared to human raters. N=1000)	Checking Twice Precision / Recall
Just Name	75.08%	87.99% / 58%
Just Description	70.31%	76.58% / 72.60%
Just Category	74.60%	86.1% / 69.7%
Just Class	70.50%	87.76 / 58.0%
All Fields	78.10%	91.72% / 65.2%
Just Color		65% / 25%
Decision Tree + **Color**		95.70% / 75.9%

*made up stats

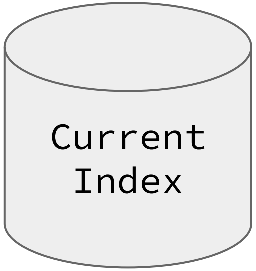
Add Color to decision tree explains more of the problem

We haven’t done the “hard work” of search

- How will we use color if it’s in the index?
- How will we balance it against other factors?
- How do we map colors in query to colors in document? Query understanding? Extraction? Etc etc?

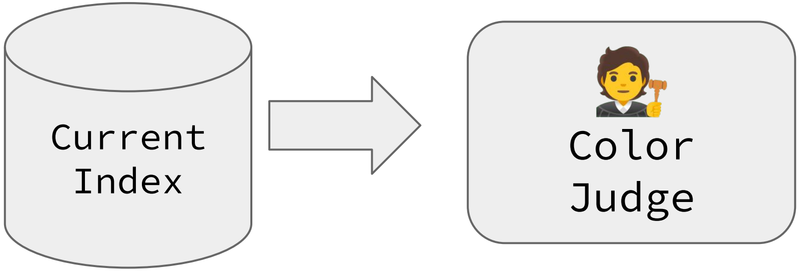
LLM Driven Relevance Engineering

Leading to LLM Driven Relevance Engineering



Naive search
solution

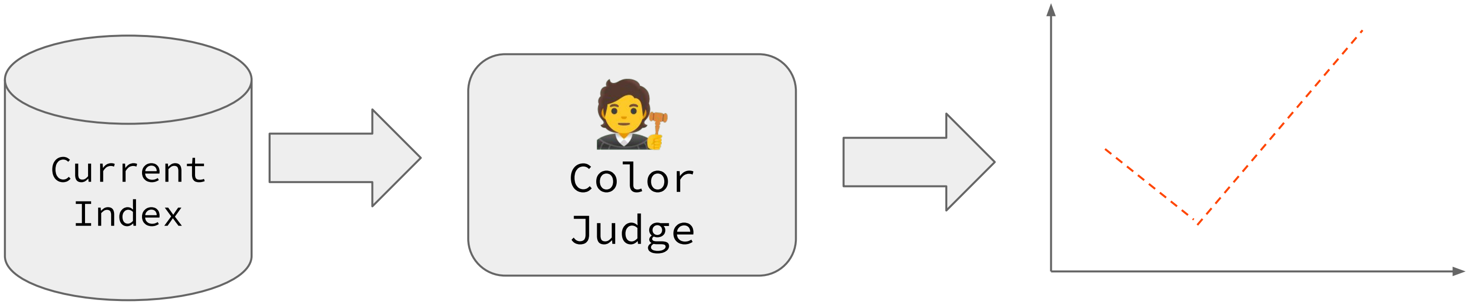
Leading to LLM Driven Relevance Engineering



Naive solution

Add new judge ~attribute

Leading to LLM Driven Relevance Engineering

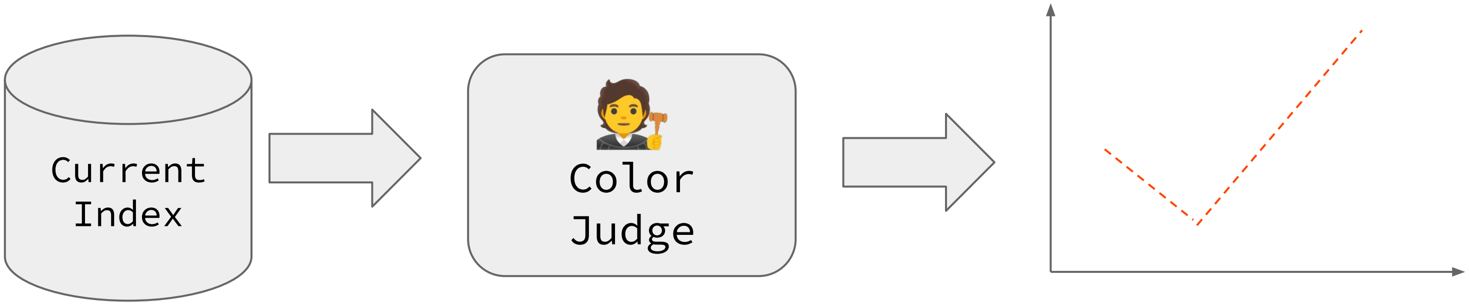


Naive solution

Add new judge ~attribute

Observe impact to **agreement** against humans in ensemble model

Leading to LLM Driven Relevance Engineering

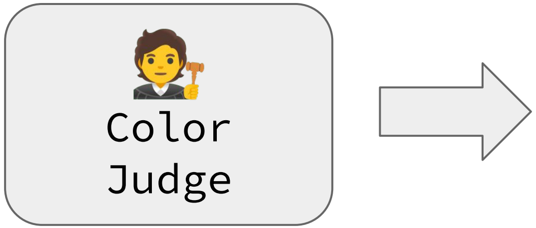


Naive
solution

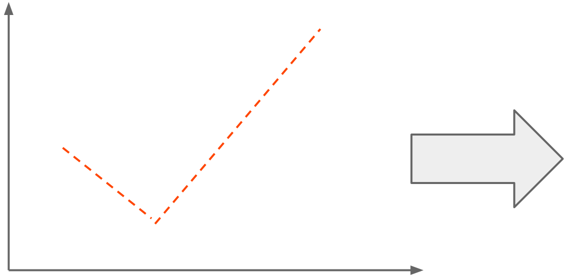
Add new judge
~attribute

(LLM Judge
Ensemble
explains more
of the **human
labels**)

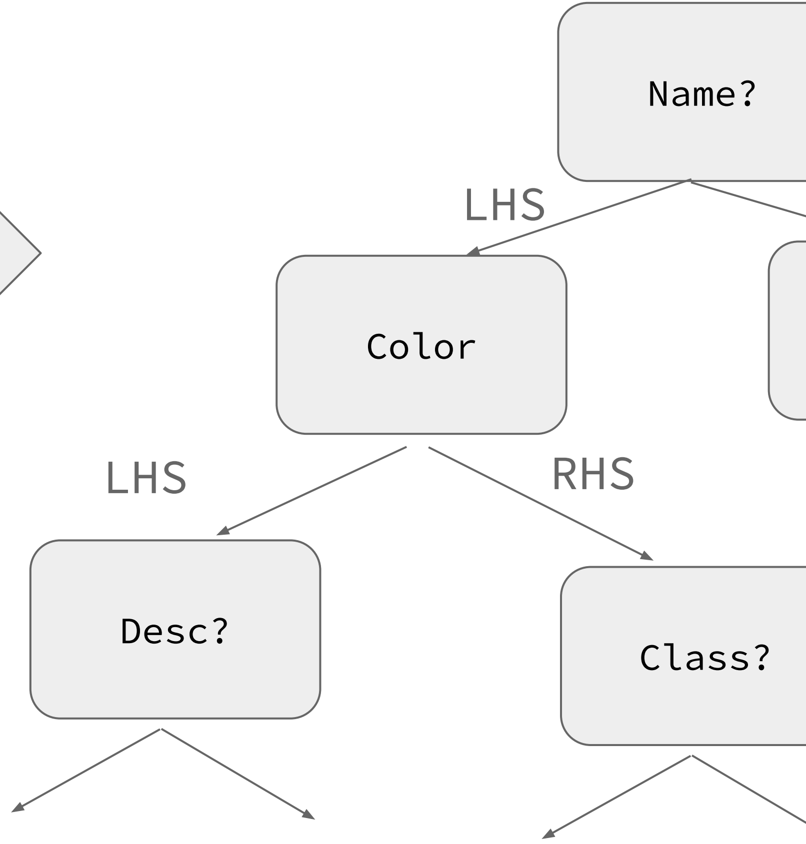
Finally seeing problem structure…



Add new judge
~attribute

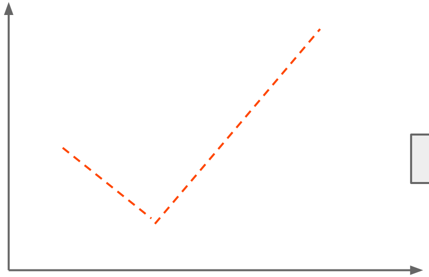


Observe impact
to **agreement**
against humans

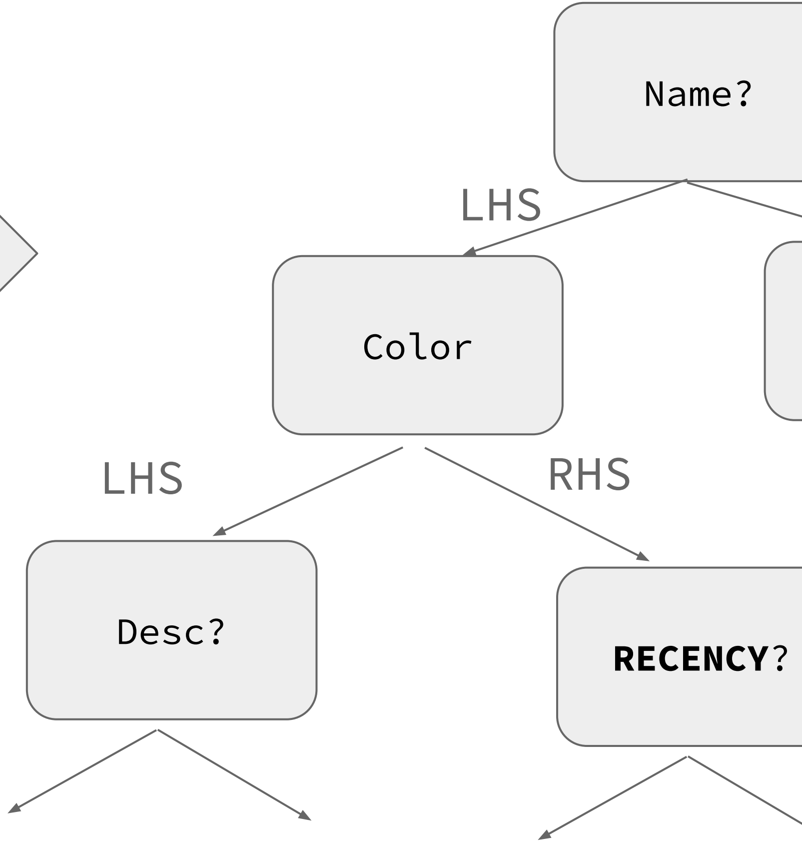


To the next thing

Add new judge
~attribute



Observe impact
to **agreement**
against humans



Questions?


Qapla’
Cheat at Sear
w/ LLMs
be-code your way to Relevant Search
maven
DOUG
TURNBULL