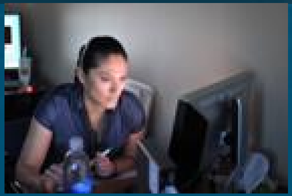
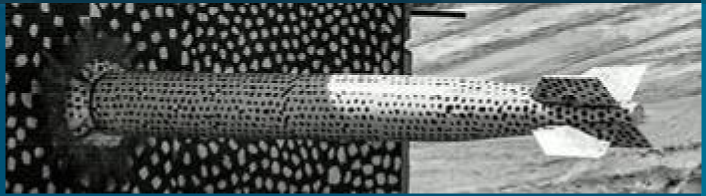
SAND2019-10363C

Sandia National Laboratories

Configuring Recommendations for Personalized Search at Sandia National Laboratories





PRESENTED BY

Clay Pryor and Ryan Cooper

SAND2019-10363 C

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Powered by Lucidworks

ACTIVATE

THE SEARCH AND AI CONFERENCE

Personalized Search at Sandia National Laboratories

CLAYTON PRYOR

RYAN COOPER

ABSTRACT

In the scope of enterprise search, the assumed preference of each user is the number of times that they have previously clicked on pages, an observed weight. This weight is then used to co-cluster (associate) with other users to make predictions about what pages they will be most likely to find useful based on their previous click history.

This presentation will describe how we configured personalized search in days, not weeks, months, or even years. We will review the configuration process from data gathering and model building to the query configuration used to return personalized results to enterprise search customers. We will share results and interesting observations as well.

Agenda

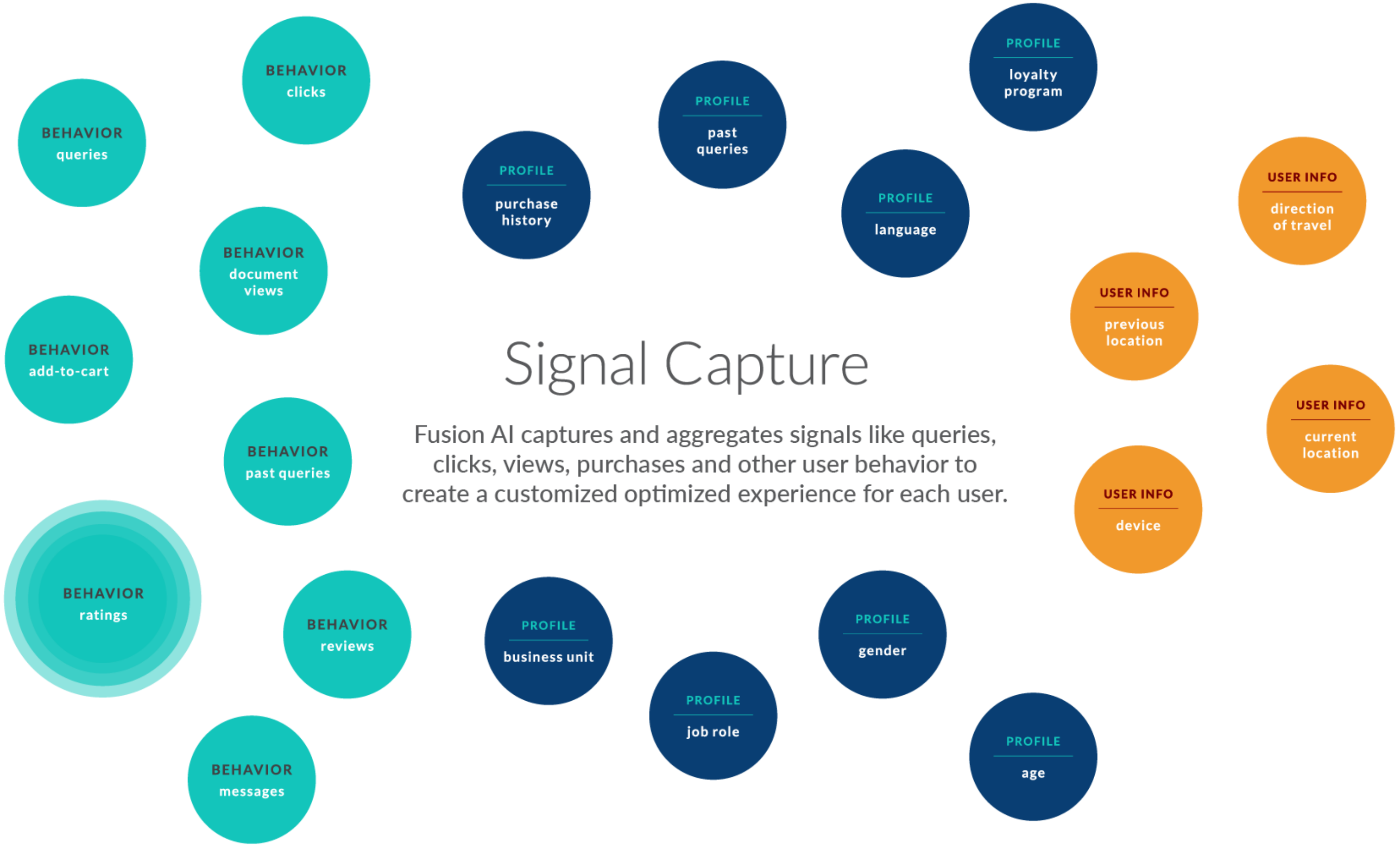
- What is Personalization?
- What is Personalized Search?
- Why does it matter in an Enterprise Search environment?

- How we accomplished data-driven personalization natively within Fusion
- Fusion configurations
- Examples
- Observations and Considerations

- Next steps

What is Personalization?

SIGNAL CAPTURE



Signal Capture

Fusion AI captures and aggregates signals like queries, clicks, views, purchases and other user behavior to create a customized optimized experience for each user.

What is Personalized Search?

Personalized search refers to search experiences that are tailored specifically to an individual's interests by incorporating information about the individual beyond the specific query provided.

Factors that could be used to influence personalized search include:

- Query History
- Click History
- Location
- Social Media
- HR Data (if acceptable)
- Organizational Data

Why does it matter in an Enterprise Search environment?

Google does it – personally, I don't like this answer ☺

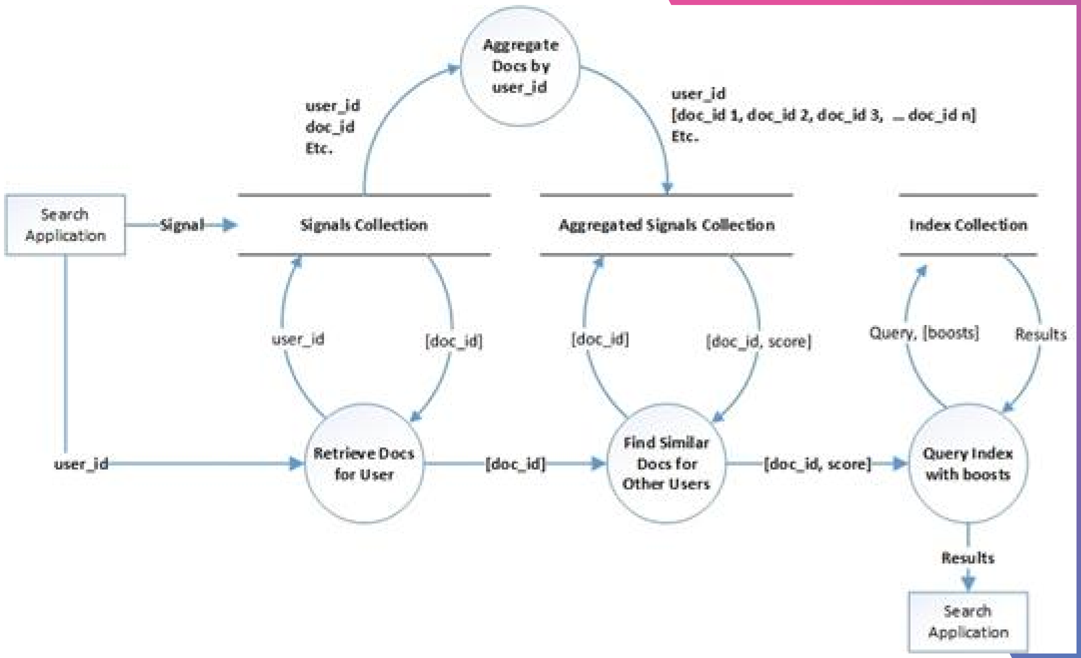
We can provide you with results that are more useful to you than the standard results

We can predict what you might be interested in without you even asking

Perhaps we can improve your safety and security

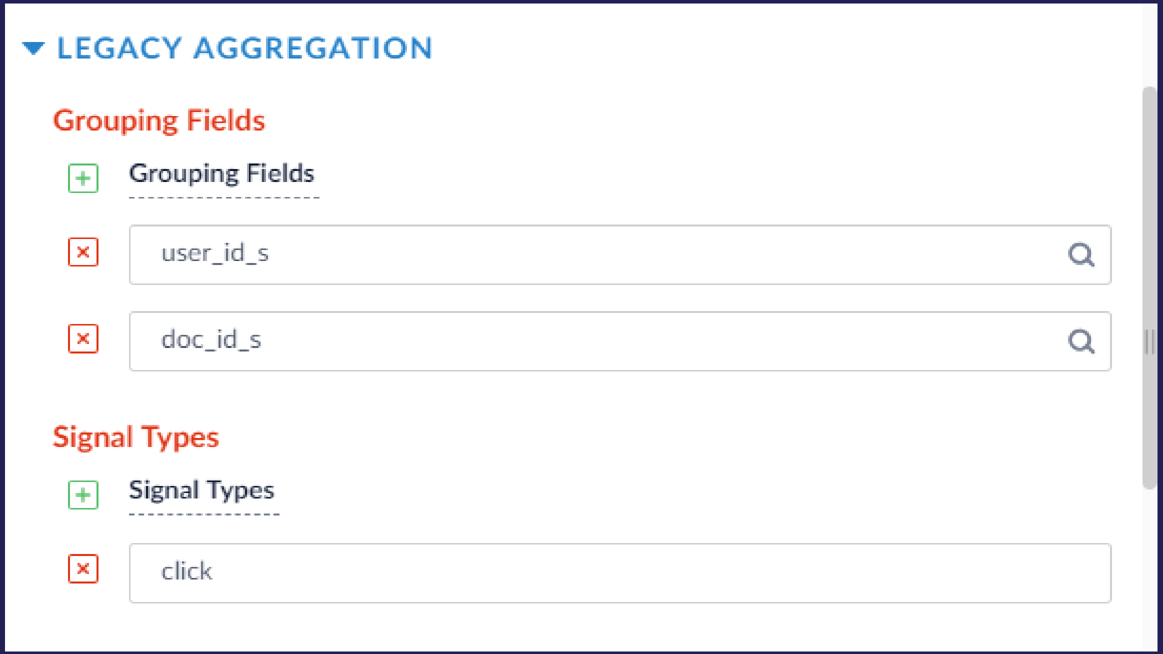
How we accomplished data-driven personalization natively within Fusion

1. Capture Signals
2. Aggregate Signals
3. Create a user-weighted documents collection
4. Train ALS Recommender Model
5. Generate “Items for Users” Recommendations
6. Incorporate Recommendations into Query Pipeline to Influence Results



Configurations of a sample personalized search pipeline within Fusion

Create a user-weighted documents collection
Job: Signals Aggregation



Train ALS Recommender Model
Job: ALS Recommender

* Spark Job ID

SNL_cf_test

The ID for this Spark job. Used in the API to reference this job. Allowed characters: a-z, A

Number of User Recommendations to Compute

100

Batch compute and store this many item recommendations per user

Exclude from Delete Filter

If the 'Delete Old Recommendations' flag is enabled, then use this query filter to identify

Number of Users to Recommend to each Item

10

Batch compute and store this many user recommendations per item

Maximum Training Iterations

10

Maximum number of iterations to use when learning the matrix decomposition

Number of Item Similarites to Compute

10

Batch compute and store this many item similarities per item

Implicit Preferences ☑

Delete Old Recommendations ☑

▼ TRAINING DATA SETTINGS

Training Data Filter Query

:

Solr query to filter training data (e.g. downsampling or selecting based on min. pref values)

Training Data Filter By Popular Items

5

Items must have at least this # of unique users interacting with it to go into the sample

Training Data Sampling Fraction

1

Downsample preferences for items (bounded to at least 2) by this fraction

Training Collection User Id Field

user_id_s

Solr field name containing stored user ids

Training Collection Item Id Field

doc_id_s

Solr field name containing stored item ids

Recommender Rank

100

Number of user/item factors in the recommender decomposition (or starting guess for it, if doing parameter grid se...

Grid Search Width

1

Parameter grid search to be done centered around initial parameter guesses, exponential step size, this number of s...

Implicit Preference Confidence

50

Confidence weight to give the implicit preferences (or starting guess, if doing parameter grid search)

Initial Lambda

0.01

Smoothing parameter to avoid overfitting (or starting guess, if doing parameter grid search). Slightly larger value nee...

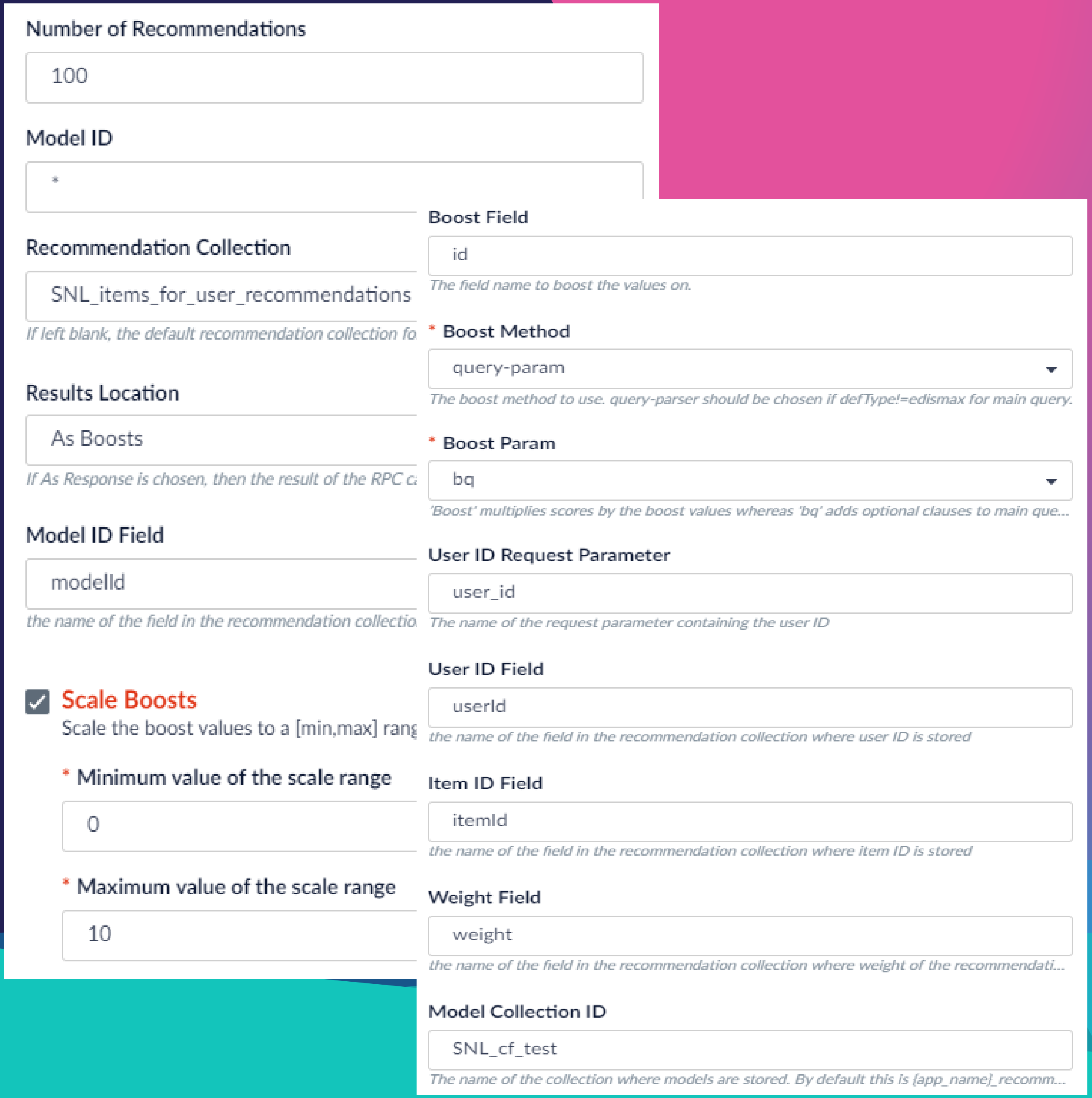
Random Seed

13

Pseudorandom determinism fixed by keeping this seed constant

Generate "Items for Users" using the resulting collection from ALS model

Query Pipeline Stage: Recommend Items for User



Examples

Results - Example – "Anonymous" v. Frequent Conference Goer

conference | Choose Sort Field | Display Fields | Parameters (2) | URI

non-personalized_pipeline

VACT Conference Rooms

title: VACT *Conference* Rooms

content: CRS, Meetings, *Conference* Rooms, *Conference* Calls, *conferance* call, teleconference, teleconferencing, *Conference* Room Scheduler

Score: 25.227148 show fields

FIN002 Obtain Approval to Attend a Conference or Sponsor or Host an Event Policy

title: FIN002 Obtain Approval to Attend a *Conference* or Sponsor or Host an Event Policy

content: FIN002 Obtain Approval to Attend a *Conference* or Sponsor or Host an Event Policy Sandia manages *conference* related decisions

Score: 23.739283 show fields

Attending a Conference

title: Attending a *Conference*

content:). Share Follow Attending a *Conference* It looks like your browser does not have JavaScript enabled. Please turn on JavaScript and try

Score: 23.601522 show fields

1 2 3 4 5 6 7 8 ...3701 Next

1-10 of 37,007 docs (28 ms, max-score 25.227148)

personalized_pipeline

Attending a Conference

title: Attending a *Conference*

content:). Share Follow Attending a *Conference* It looks like your browser does not have JavaScript enabled. Please turn on JavaScript and try

Score: 32.163067 show fields

↑ Rank +2

Expense Report Plus

title: Expense Report Plus

content: Travel, WebER, Reimbursements, Foreign Travel, International Travel, Employee Expense Reporting, *Conferences*, ETA, ET, Expense

Score: 27.81516 show fields

↑ Rank +4

VACT Conference Rooms

title: VACT *Conference* Rooms

content: CRS, Meetings, *Conference* Rooms, *Conference* Calls, *conferance* call, teleconference, teleconferencing, *Conference* Room Scheduler

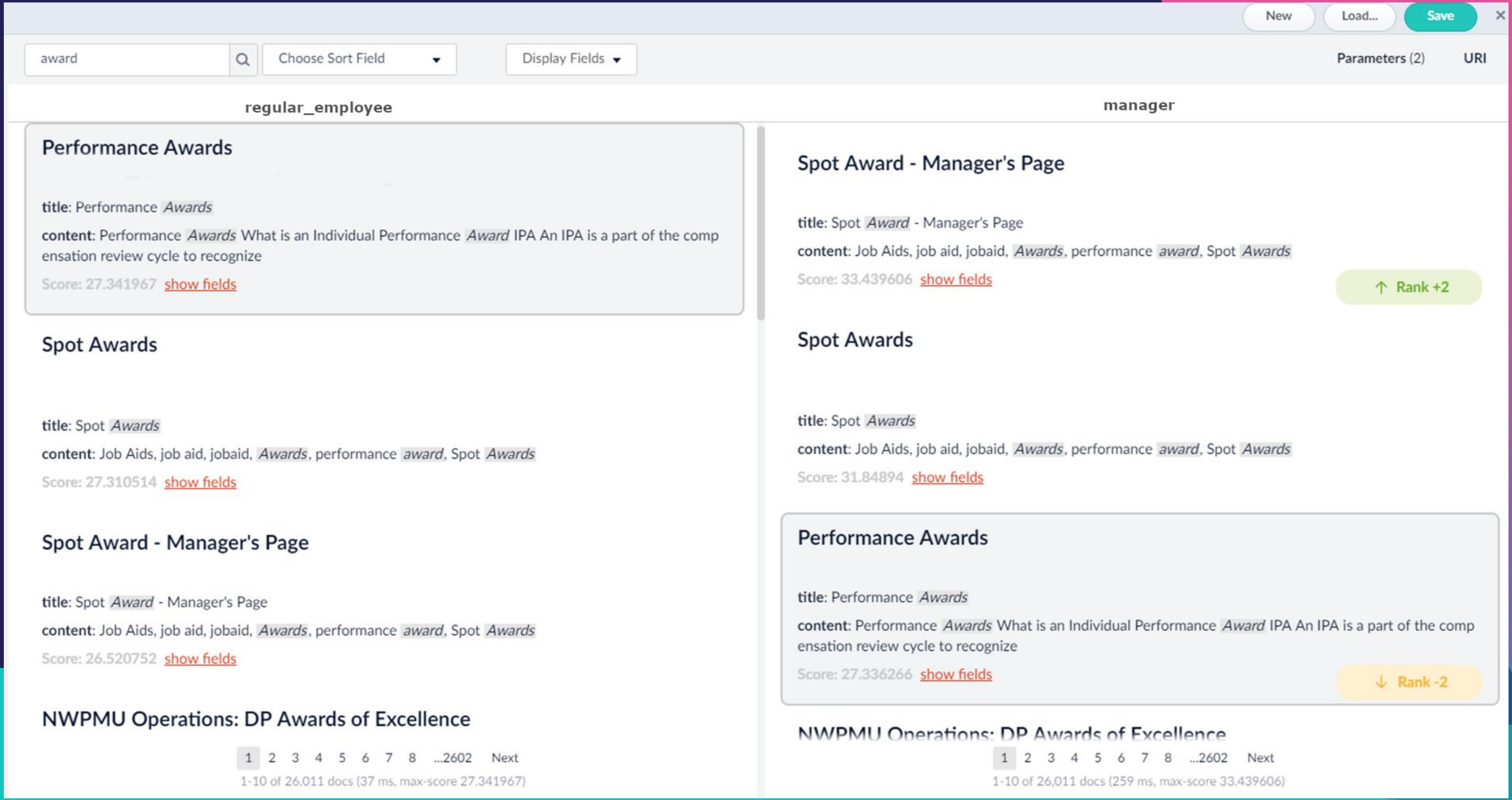
Score: 25.227148 show fields

↓ Rank -2

1 2 3 4 5 6 7 8 ...3701 Next

1-10 of 37,007 docs (51 ms, max-score 32.163067)

Results - Example – Regular Employee v. Manager



Observations

- It worked! The results change when personalization is enabled.
- This personalization is not based solely on the current user's click history but what other individuals with similar interests clicked on
- This approach produces "inferences" of what the current individual might be interested in
- Similar types of users had similar recommendations
  - Frequent Conference Goers
  - Managers
- Level of personalized results is configurable
- A user with no click history is given standard search (no personalization).

Considerations

- The model generates a query-agnostic list of recommended documents and scores.
- Only documents that pass the query parsing stage will receive a boost, implying that irrelevant documents to the query will be filtered out, and thus not boosted.
- However, pages will be boosted whenever they pass the query parser.
  - i.e. if https://www.example.com/ appeared in the queries “foo” and “bar”, it will be boosted in both.

Next Steps

- We have a policy not to introduce changes into our search application until we evaluate them
- Changes must not hurt the search results and, hopefully, they should improve the search results
- Compare to Golden Standard - We have a tool that will evaluate changes in search results based on how well they match what experts list as the most desirable results for selected queries.
- Golden Standard does not work when results change depending on who is currently using the system!
- Must consider alternatives such as
  - Focus Groups
  - A/B Experiments

References

1. https://doc.lucidworks.com/fusion-ai/4.1/user-guide/boosting/index.html
2. https://en.wikipedia.org/wiki/Implicit_data_collection
3. https://en.wikipedia.org/wiki/Recommender_system
4. https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe
5. https://lucidworks.com/2017/08/24/machine-learning-model-training-and-prediction-using-lucidworks-fusion/

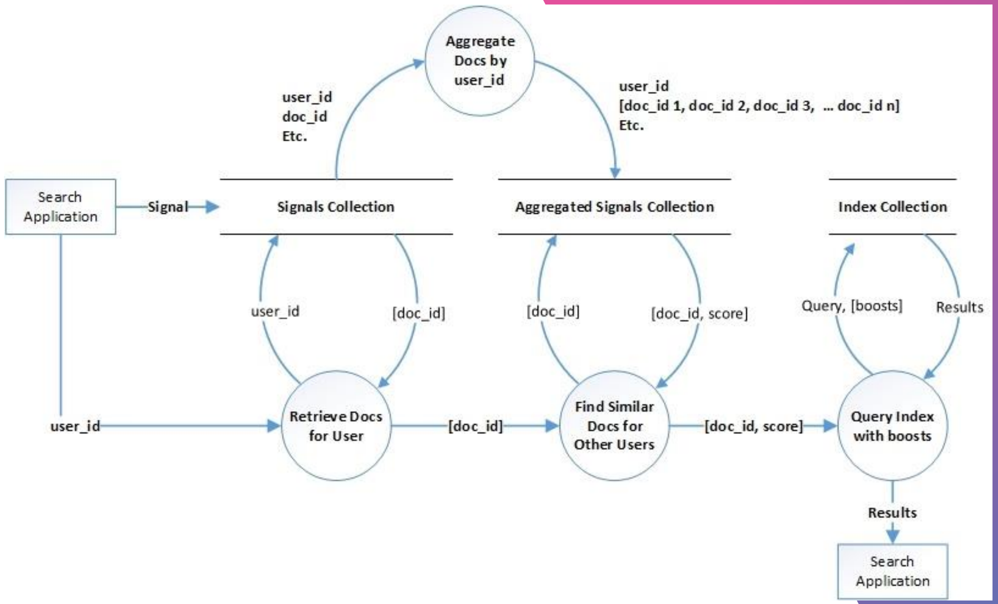
THANK YOU

Previous Approach

Previously, Clay had demonstrated personalization by using custom solr sub-queries to lookup and do collaborative filtering on-the-fly from within the query pipeline.

In this approach, there are three mechanisms occurring:

- $f : user \rightarrow \{page_0, \ldots, page_n\}$
  - Gather user click history, a set of *n* pages
- $g : \{page_0, \ldots, page_n\} \rightarrow \{(page_0, score_0), \ldots, (page_m, score_m)\}$
  - Find related documents based on collaborative-filtering-like solr query.
- $h : (query, \{(page_0, score_0), \ldots, (page_m, score_m)\}) \rightarrow (page_0, \ldots, page_n)$
  - Query the index with additional boosts from the related documents.



Approach

Fusion provides an ALS (alternating least squares) collaborative filtering job that trains a model to recommend pages for users.

By pretraining this model, it essentially accomplishes the functionality of both *f* and *g* from the previous approach, while also eliminating the runtime computation.

It also simplifies the mechanism.

- $a : user \rightarrow \{(page_0, score_0), \ldots, (page_m, score_m)\}$
  - Evaluate the model with the user to get a list of recommended pages and scores
  - This is essentially a composition of *f* and *g* from the previous approach
- $\mathrm{b} : (query, \{(page_0, score_0), \ldots, (page_m, score_m)\}) \rightarrow (page_0, \ldots, page_n)$
  - Query the index with additional boosts from the recommended pages.