Solr Payloads and Joins

Kevin Watters
KMW Technology
Berlin Buzzword 2021/06/16

Who am I?

- Founder of KMW Technology, in operation since 2010
- Based in Boston
- Focused on Solr and Elasticsearch based applications.
- Provide training and search cluster architecture review.
- Custom application development
- Open Source supporters, contributors and committers!

KMW TECHNOLOGY

Solr Payloads - Solr 9.x

What are payloads?

Each term in document can have a position (.pos).

Terms have positions.

Positions have Offsets

Offsets Point to payload data.

What can you use them for?

The payload_check and payload_score query parsers can use payload values for matching and scoring

What did we improve?

Previously, the payload_check query parser could only perform an equals operation, **now it supports inequalities!**

KMW TECHNOLOGY

Payload Delimited Fields

Includes the DelimitedPayloadTokenFilterFactory

Delimited payload factor specifies the “encoder” for the byte array:

- Int - integers
- Float - floating point values
- Identity/string - pure byte comparison (for eq) (string comparison for inequalities)

```
<fieldType name="delimited_payloads_float" class="solr.TextField" indexed="true" stored="false">
   <analyzer>
     <tokenizer name="whitespace"/>
     <filter encoder="float" name="delimitedPayload"/>
   </analyzer>
 </fieldType>
```

```
Data Input Format:

Field:   value|payload  value2|payload2   value3|payload3  ...
```

KMW TECHNOLOGY

Payload Query Parsers

Payload score - used to look up a payload to be included in a function query for scoring. (numeric fields only)

Use the max value from a payload from the term that matched the query.

```
{!payload_score f=”myfield_dpf” v=”querystring” func=”max”}
```

Payload check - used to validate that a payload matches the criteria for a given term to match a query. (previously only equals was supported.)

Searching for a train, where it has a payload of “noun”

```
{!payload_check f=”words_dps” v=”train” “payloads=NOUN”}
```

(New!) Searching for a train where the payload is greater than 0.75

```
{!payload_check f=”words_dps” v=”train’ payloads=”0.75” op=”gt”}
```

KMW TECHNOLOGY

Example Use Case - Searching for Image Classifications

Machine learned models and AI can extract and classify information. Typically a classifier will produce a label and a confidence score that the label was correctly identified.

The goal here is to index the output layer of a document being passed through a neural network.

Add to the document the labels and confidence scores that come from that network output layer.

For image classification, in addition to the label and confidence, a bounding box or region of interest (ROI) might also be returned as part of the network output layer.

The use case presented here is searching through the results of image classification using the popular open source neural networks VGG16 and Yolo.

KMW TECHNOLOGY

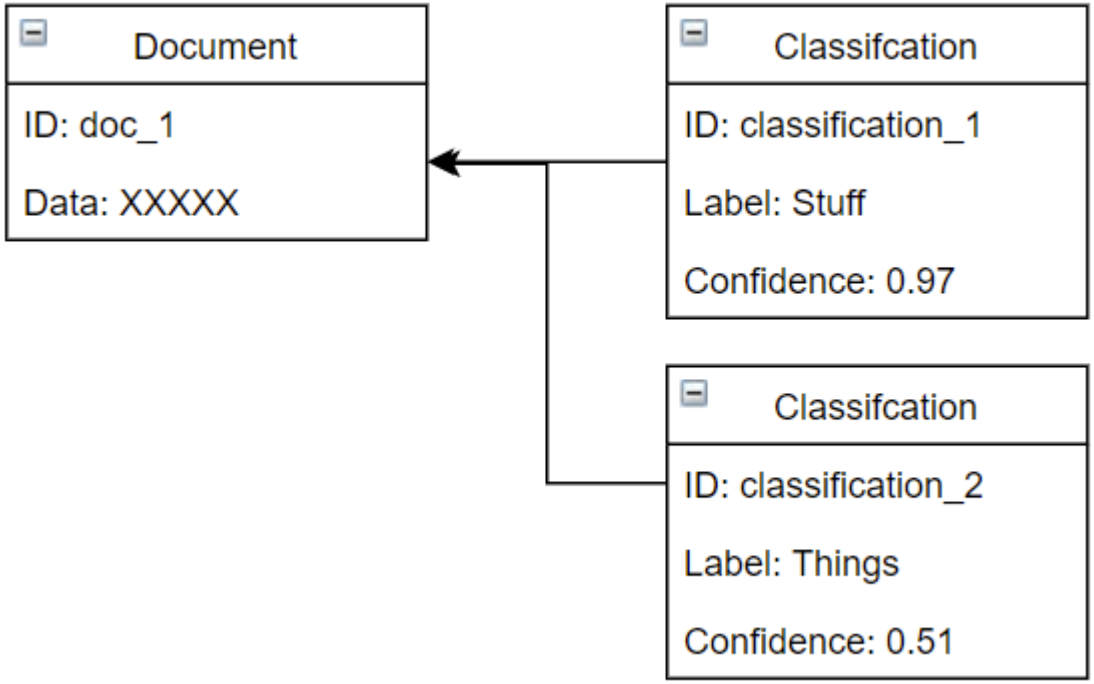
Example Document w/Payloads

Based on VGG16 & Yolo output, we can create a document like the following. We index the labels with their confidence levels, additional we index position and size of the bounding boxes.

```
{
  "url":["http://phobos/cocodataset-train2017/train2017/000000006053.jpg"],
  "max_vgg16_confidence_d":0.7800717949867249,
  "ski":[0.7800717949867249],
  "vgg16_dpfs":["ski|0.780072",
                "alp|0.194378",
                "snowmobile|0.005578",
                "ski_mask|0.003939",
                "shovel|0.003189",
                "dogsled|0.002374",
                "bobsled|0.001709"],
  "yolo_count_dpfs":["PERSON|1"],
  "yolo_x_dpfs":["PERSON|183.0"],
  "yolo_y_dpfs":["PERSON|289.5"],
  "yolo_size_dpfs":["PERSON|27666.0"],
  "yolo_dpfs":["PERSON|0.818751"]
}
```

KMW TECHNOLOGY

Example Classification Data Model



KMW TECHNOLOGY

Approach 1 - Filter at Index time

Only tag documents with labels that are above a certain threshold.

At query time, search the field, knowing that only items over the threshold is in it.

Pro : Very simple to implement
Pro : Very fast queries
Pro : Small index
Con : Can’t change your threshold at query time! (more flexibility at query time requires additional fields for low, medium, high confidence versions of the labels.)

KMW TECHNOLOGY

Approach 1 - Details

Example Document (Single multi value field with labels)

```
{
  "id":"doc_78375",
  "high_confidence_ss":  ["cat","dog", ...]
}
```

Example Query

```
high_confidence_ss:cat
```

KMW TECHNOLOGY

Approach 2 - Dynamic fields for labels

Use a numeric dynamic field with a field name equal to the label, and the value in the field is the confidence.

Index each label as a new field, the value in the field is the confidence level.

Pros: Can change the threshold by using a range searches

Cons: Potential explosion in the number of fields (one field per label)
Cons: Labels might not be known ahead of time.

KMW TECHNOLOGY

Approach 2 - Details

Example Document

```
{
  "id":"doc_20295",
  "cat_fs":[0.5059256],
  "dog_fs":[0.8075591],
  "person_fs":[0.6512147],
  …
}
```

Example Query

```
dog_fs:[0.75 TO *]
```

KMW TECHNOLOGY

Approach 3 - Use a join query

Create children documents that contain the label and the confidence, join them back to the parent record to filter.

Pro: Full flexibility in filtering logic with the join query.

Con: Queries are much slower/more expensive.
Con: Children/Classification documents add to overall index document count
Con: Requires custom routing on the join key which makes sharding more difficult.

KMW TECHNOLOGY

Approach 3 - Details

Example Parent Document

```
{
  "id":"doc_729413"
}
```

Example Child/Classification Document

```
{
  "id":"doc_729413_5",
  "parent_id_s":"doc_729413",
  "label_s":"cat",
  "confidence_f":0.81850449
}
```

One for each classification, ~ 50 per document for this example)

Example Query

```
{!join to=”id” from=”parent_id_s” v=”+label:foo +confidence:[0.75 TO *]”}
```

Approach 4 - Use payloads and inequality queries

Extend the payload_check query parser to support an operation while matching payloads. The operation could be greater than, less than or equal to. (gt, lt, gte, lte, eq)

Encode the confidence score as a floating point number in the payload associated with each label given to the document.

Index the values in a float payload delimited field.

Use the new “op” local param on the payload_check query parser.

KMW TECHNOLOGY

Approach 4 - Details

Example Document

```
{
  "id":"doc_doc_333114",
  "classification_dpfs":["cat|0.7949082", "dog|0.22995031", "person|0.66999483", ...]
}
```

Example Query

```
{!payload_check f="classification_dpfs" payloads="0.75" op="gt" v="cat"}
```

KMW TECHNOLOGY

Benchmarking Notes

For indexing the document, a single threaded simple java application sending documents to solr was used to build a representative index of documents with their classifications.

Dataset consisted of documents with an average of 50 classifications. Each classification had a confidence score between 0.0 and 1.0 and a randomly selected label.

There were 10,000 unique labels in the generated benchmark test dataset.

At query time, the filter caches and query caches were disabled to avoid caching. Queries were executed serially on a single thread one for each label in the index.

KMW TECHNOLOGY

Indexing Benchmarks

Indexing rates and memory usage show the expense of many fields on a document compared to the payload data.

	Approach 1 (index time)	Approach 2 (field per label)	Approach 3 (child doc join)	Approach 4 (payloads)
Docs/Second	11,761	209	409	3,566
Index Size (kb)	219	906	2,610	1,220
Memory (bytes)	1,772	1,361,108	5,556	1,460

KMW TECHNOLOGY

Query Benchmarks

Join queries were so slow, the benchmark was stopped after 100 queries.

	Approach 1 (index time)	Approach 2 (field per label)	Approach 3 (child doc join)	Approach 4 (payloads)
Num queries	10,000	10,000	100*	10,000
Avg Response (ms)	1	2	1,803	3
Min (ms)	1	1	1,656	2
Max (ms)	6	22	2,092	47
std dev	0.55	0.79	88.28	1.55
Queries/sec	636	347.9	0.557	254
Avg Result Size	9,861	26,461	1,499	3,826

KMW TECHNOLOGY

Demo Image Processing Pipeline and Example Index

- Indexed COCO dataset. Each image was run through OpenCV and Deeplearning4J
- VGG16 : Image classification, 1000 labels with confidence score
- Yolo: Classification and Localization 80 object labels, with confidence score and bounding box information
- DNN Face Detection: Detect faces with confidence and bounding boxes
- Blur Detection

KMW TECHNOLOGY

Example Document w/Payloads

Based on VGG16 & Yolo output, we can create a document like the following. We index the labels with their confidence levels, additional we index position and size of the bounding boxes.

```
{
  "url":["http://phobos/cocodataset-train2017/train2017/000000006053.jpg"],
  "max_vgg16_confidence_d":0.7800717949867249,
  "ski":[0.7800717949867249],
  "vgg16_dpfs":["ski|0.780072",
                "alp|0.194378",
                "snowmobile|0.005578",
                "ski_mask|0.003939",
                "shovel|0.003189",
                "dogsled|0.002374",
                "bobsled|0.001709"],
  "yolo_count_dpfs":["PERSON|1"],
  "yolo_x_dpfs":["PERSON|183.0"],
  "yolo_y_dpfs":["PERSON|289.5"],
  "yolo_size_dpfs":["PERSON|27666.0"],
  "yolo_dpfs":["PERSON|0.818751"]
}
```

KMW TECHNOLOGY

Image Search Example / Demo

Image classification has a label and a confidence score.

“Show me at least 2 people and a garbage truck”

+{!payload_check f="yolo_count_dpfs" op="gte" payloads="2"}PERSON AND +{!payload_check f="vgg16_dpfs" op="gte" payloads="0.70"}garbage_truck

KMW TECHNOLOGY

Next Steps and Future Considerations

Refactor and cleanup the Payload codec (encoders / decoders) in Lucene

Vector Matching? Cosine similarity and others

Find Similar images based on neural network classifications

NLP / NLU query parser

KMW TECHNOLOGY

Questions & More Info

More Info: https://kmwllc.com/index.php/2021/06/12/solr-payload-inequalities/

LUCENE-9659 Added the updates to the SpanPayloadCheckQuery
SOLR-14787 Added the updates to the payload_check query parser.

Contributors:

Kevin Watters
Gus Heck
David Smiley

KMW TECHNOLOGY