**Basic Scoring Steps**

1. Acquire identifiers for unprocessed data
   1. How to do this? We can’t do this from the API itself, so in the end we might need direct database access for this purpose.
   2. *curl -X GET "https://open.ompnetwork.org/api/sessions?start=0&limit=100&live=0&createdAfter=<DATE\_IN\_UNIX\_TIMESTAMP\_FORMAT>" -H "accept: application/json"*
   3. Might need to page through this, depending on the number of sessions
2. Acquire unprocessed data
   1. *Curl -X GET “*[*https://open.ompnetwork.org/api/session/38295/captions*](https://open.ompnetwork.org/api/session/38295/captions)*” -H “accept: application/json”*
   2. Might need to page through this, depending on the number of captions
3. Categorize new data
   1. Leverage example provided by Anna in the Jupyter notebook for tagging *sessions* with topics
   2. Use the part of (a) which assigns topics on a caption basis to save the top N topics per caption (along with scores of relevancy?)
   3. How do we make a relevancy score? I think there’s some measure like this used inside the categorization from the previous Delta team, let’s look at that and see if that’s good to use.
4. Write categorized data back to the database
   1. sessions table:
      1. Fields: (session\_id, int), (created\_at, timestamp)
   2. session\_captions table:
      1. Fields: (session\_id, int), (caption\_id, int), (topic\_id, int), (total\_seconds, int)
   3. topics table:
      1. Fields: (topic\_id, int), (topic\_text, text)
   4. session\_analytics view:
      1. Join sessions to sessions\_captions; group by week, topic, sum(total\_seconds) where topic in x
   5. topics\_analytics view:
      1. sessions\_captions; group by topics sum(total\_seconds) where topic in x