What do we want and need for building a ranker/search system?

Want?

A repeatable, efficient, and sufficient way of testing the quality of our _______ system. The blank can be

1. E2E search system 2.Retriever

3. Ranker

Metrics:

1. Quantitative - Qualitative - User Search Experience Parity
2. Measure a system using math - Quantitative Performance
3. Measure a system using internal Human in Loop testing - Smoke tests - SME involved testing
4. Measure a system's impact on the User's search experience - Implicit user signals
5. A way to make sure all the three signals correlate regarding the performance of a system -> if the nDCG is high - Smoke test should be positive - User's search experience should improve. This is the ideal state
6. In a world where the dataset suffers from missing judgement labels, relying on metrics of just one kind might not be enough

Need?

What is our ranker system?

Re-orders the retrieved documents to display the best possible documents to the user.

Key Assumptions

1. **Ranking behaviour is learnable and generalizable**
    a. Ranker Options:
        - Heuristic Ranking
        - AI - LambdaMART - BERT etc
    b. The ranker training data
        - Is reflective of the real world
        - Is sufficient

    - The ranker performance / learning capacity is measurable
    - Adds measurable and meaningful value

  c. Comments

    - Most of the key assumptions have been proven true for other general english based IR (Information Retrieval) domains.
    - Many of the metrics used for measuring ranking quality suffer from the lack of a complete set of judgements. The universe of perfect answers for a given question keeps changing, and isn't easy to construct.
    - These issues challenge multiple key assumptions that need to be true for a ranker to be trained and deployed in a way that can generate measurable performance improvements for both qualitative and USX dimensions.
    -

2. **How to measure the quality of the Ranker and the data?**

   a. **Ranker**

    - **Assume a perfect world**
    - Our training data is good and sufficient
    - The training and test data are derived from the same distribution
    - The source data distribution and the sampling is optimal
    - Info shown to the user conveys the magnitude of relevance of the card
        - If we retrieve a highly relevant decision the user will understand the same and click on it
    - **Questions to answer**
        - What is the scope of our ranker (John's Son's)
            - Do we rank only issue based similarity
            - Do we rank multiple latent/meta data signals like popularity
            - Do we need a ranker?
        - Quantity of signals available beyond the retriever to make meaningful and performant ranker
        - Is ranking needed for all kinds of queries?
            - Boolean Search Engines typically don't have rankers
            - KW queries probably don't need it
            - If a ranker adds significant value, the user might want it to be run on all types of queries
        - Is measuring metrics at a small k like 5 sufficient for our system

TODO: Fill In the blanks 1

1. Does it help us surpass the missing labels problem iv. What portion of the universe of answers needs to be known
   v. When should a ranking metric be 1 or 100% for our system?

1. In a case where the entire universe is known and labelled 2. In a case where it isn't vi. What do we need to say we say R1 is better than R2 vii. Ways to surpass the missing label - Section above viii. What features should be included into the

ranker - What should our

ranker be?

1. Multiple rankers ? 2. Heuristic? 3. Can a text based ranker provide more value than any possible

retriever improvements? ix. Impacts of training using incomplete data

x. Should the query and label distribution of training data represent the real

world xi. Should we include an approximation of our metrics in the loss function

when training a ranker

1. Like nDCG for LambdaMART xii. Should the metrics for training and test be measured at the same k? xiii. Could a perfect ranker provide perfect USX?

1. If our ranker is perfect, would the user be happy with our page 1 of

results? xiv. Ranker as a classifier

1. Have a classifier trained on existing data to predict possible label

b. Testing Data

- Scope of testing?
  - Test E2E & Ranker in the same testset?
  - Question type coverage
  - Should there be multiple types of questions and intents in the same testset
  - Should the development of a query be represented
  - Should there be numerous smaller testsets
  - Frequency of updates
  - Should it be sampled from the same distribution
    - Sample from training dataset - Save/Hide a portion of the training data ii. Mix of training and user data - Sample 50% from the training data, and 50% from user queries (curated/raw/ representing user query intent distribution)

c. Characteristics of the test set

- i. Size - Num of question and Num answers per question
- ii. Label distribution
- iii. Could question chosen/hidden from training set be paraphrased versions of the training question?
- v. Should it contain legal signals similar to that of the training set
- How much does labelling cost
- Is freezing the universe for test set for legal signal dependant ranker valid

d. What is the best way of collecting labels

- Label all retrieved documents at retrieval size - As these can be the only documents seen by the ranker
  - 1. Tightly coupled with retriever - Theoretically the amount of effort needed should converge to the end quickly, but if our retriever isn't perfect/ any

retriever improvements might slow down the convergence

        - 2. One time expense
    - Capture the perfect universe of answers - articulated
        - Find “all” the good answers
        - Reasonably freezable - Everything else can be confidently irrelevant/topical - hence a 0
        - Costly - Needs resources outside of ROSS iii. Continuous/Incremental labelling
        - TODO FIll the blanks two?
            - 1. Choose a small @k for metrics 2. Label new data points on every experiment run 3. Measure complete metrics instead of condensed 4. Tied to retriever - Theoretically the amount of effort needed should converge to the end quickly, but if our retriever isn’t perfect/ any retriever improvements might slow down the convergence iv. Effort needed in process 2 and 3 is proportional to the size of test dataset h. Viable proportion of missing labels? i. Smaller testset - get metrics - error analysis to see if test set serves intended
        - purpose - Iterate j.

- e. **Relevance Interpretation**
    - Only text based ranker
        - Relevance signifies pure textual similarity to the question
    - b. Multiple signals based ranker
        - Relevance is implicitly associated with all these signals, and hence needs to be made explicit to the labelled, and validated
        - 1. If an answer gets score 3 - Does it mean the it has all the signals
        - 2. Train a ranker with different combinations of the extended feature set to evaluate correlation
        - c. Would we need multiple judges to ensure the absence of labeller proclivities
- f. Test Datum
    - i. Numerical similar to the training data ii. Qualitative - Implicit user interactions

4. Ideal Question

a. How much should it reflect real world?

i. Length of the query

ii. KWs?

iii. Intent of the query? - Different queries have different search intents, should we differentiate that in the data

iv. Level of curation

v. Inclusion of product specific information

TODO FIll in the blanks 3

1i. Do we need/want it? ii. What does it mean?

1. Rearrange words 2. Synonyms (English and Legal) iii. Can our current system handle paraphrasing? c. Ideal distribution of the types of queries d. Do the same rules, metrics, ranker rules apply for all types of queries e. Test Datum

i. Captures user's behaviour and expectations ii. Can measure the effect of training

5. Ideal Answer

- a. It needs to "answer" the question
  - i. Different intents of answers, different levels of answers or types of answering
  - ii. Do answer intent match question intents
- b. Do the chosen answers have retrieves signals?
  - i. Will it get retrieved?
  - ii. If not it should be used for recall/ retriever relevance tuning
- c. Should an answer text span reflect what is displayed to the user?
- d. What is an answer
  - i. Document ii. Context iii. Summary iv. Paragraph
  - v. Or a combination of all

ML Notes - notes If machine learning is not absolutely required for your product, don't use it until you have data.

Choose machine learning over a complex heuristic

Don't export if problems - https://developers.google.com/machine-learning/guides/rules-of-ml/#rule_9_detect_problems_before_exporting_models **if you find yourself increasing the directly optimized metric, but deciding not to launch, some objective revision may be required. -https://developers.google.com/machine-learning/guides/rules-of-ml/#rule_12_don%E2%80%99t_overthink_which_objective_you_choose_to_directly_optimize**

Moreover, no metric covers the team's ultimate concern, "where is my product going to be five years from now"?

Action items/Ideas

1. IDCG is @p -> figure out how many 3 and 2s need to exist - the distribution 2.