Family History Discovery through Search at Ancestry

Peng Jiang*
Yingrui Yang*
pjiang@ancestry.com
yyang@ancestry.com
Ancestry
San Francisco, California

Gann Bierner
Ancestry
San Francisco, California
gbierner@ancestry.com

Fengjie Alex Li
Ancestry
San Francisco, California
ali@ancestry.com

Ruhan Wang
Ancestry
San Francisco, California
rwang@ancestry.com

Azadeh Moghtaderi
Ancestry
San Francisco, California
amoghtaderi@ancestry.com

KEYWORDS

federated search, learning to rank, diversity metric, genealogy

ACM Reference Format:
Peng Jiang, Yingrui Yang, Gann Bierner, Fengjie Alex Li, Ruhan Wang, and Azadeh Moghtaderi. 2019. Family History Discovery through Search at Ancestry. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3331184.3331430

1 GENEALOGY SEARCH AT ANCESTRY: PRACTICAL CHALLENGES

At Ancestry, we apply learning to rank algorithms to a new area to assist our customers in better understanding their family history. The foundation of our service is an extensive and unique collection of billions of historical records that we have digitized and indexed. Currently, our content collection includes 20 billion historical records. The record data consists of birth records, death records, marriage records, adoption records, census records, obituary records, among many others types. It is important for us to return relevant records from diversified record types in order to assist our customers to better understand their family history.

Searching such a diverse set of content presents a number of challenges.

(1) The available information from each record type and the relevance to the query can vary greatly. When searching for a person, the search system must take into account the fact that search terms may be more valuable within some types of records than in others. For example, while a married name is not present in birth records, it is valuable in census records and death records.
(2) We need to deal with different types of features. For example, we have features defined on various fields (such as names, places, etc) indicating the comparison between queries and records. We also have features measuring the importance of each record type. It is important to properly use these features based on their properties in our ranking models.
(3) The return on investment (ROI) of each collection of records could vary. We would like to have a way to calculate ROI so as to compare different investment opportunities over multiple record collections.

In the proposal, we will describe how these challenges are addressed in Ancestry. We introduce the design of our architecture that federate the results across record types. We then propose two machine learning models to handle different feature types. Furthermore, we propose a novel information retrieval metric to measure the diversity of results. We would like to share our learning of applying information retrieval techniques to the this novel domain of genealogy. We hope to have further collaborations with researchers in the IR community.

2 GENEALOGY SEARCH AT ANCESTRY: ARCHITECTURE, FEATURES, MODELS, AND METRICS

We address each of the challenges from architecture, features, models, and metrics respectively.

ARCHITECTURE. To address the disparity issue in data sources, the Ancestry search system supports the ranking behavior by building a specialized query for each record type. A query is customized based on specific requirements for each record type. A machine learning model is then applied to retrieve top 100 (if applicable) records from each record type. Each record then has a predicted score by the record specific model. This process is referred to as *record specific search.* These predicted scores are then combined using a collator based on another machine learning model. This model predicts a weight for each record type. The final ranking score of each record is then re-scaled by its corresponding record type weight. This process is referred to as *global search.* We illustrate the architecture in Figure 1.

MODELS. We use hierarchical models to deal with different types of features. Specifically, the record specific models use binary

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '19, July 21–25, 2019, Paris, France
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331430

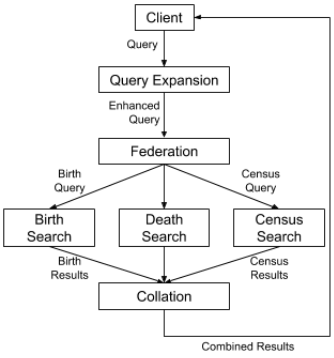

Figure 1: Example Search Process

features indicating the similarity of different fields between queries and records and predict a score for each record. The global search model uses these scores as features and predicts the importance of each record type.

We propose two ranking models based on two optimization algorithms: Coordinate ascent (CA) and Nelder–Mead (NM) respectively for each searching scenario. CA is known for its slow convergence. We therefore propose customized CA for applications where features are binary to speed up the procedure. We modify weight initialization schema provided by the library RankLib which initializes the same weight for every feature. Our formula for initializing the weight of a feature is given as follows:

$$w = \begin{cases} 0.5, & \text{if } fre\_rel = 0, fre\_irrel = 0 \\ \frac{fre\_rel}{fre\_rel + fre\_irrel} & \text{otherwise} \end{cases}$$

where fre_rel is the number of times for the feature being 1 in relevant records, and fre_irrel is the number of times for the feature being 1 in irrelevant records. The customized CA converges 10 times faster than canonical CA in offline record specific search experiments while maintaining the ranking performance.

In contrast to classical federated search methods, where coexistence of documents and chorus effect are commonly exploited, our global search is a special use case as the databases have no overlaps and the relevance scores from each list are not necessarily cooperative. We therefore treat this as a learning to rank problem that linearly combines ranked results federated across contents from various record types. In our application, we have around 10 unknown parameters (one for each record type). We adopt NM with weights initialized by rankSVM and name our method as NM-rankSVM. Although NM optimizing for listwise loss sounds computationally expensive, the low dimensionality makes it feasible. First, it learns an initial set of weights based off rankSVM on part of the training set. Then it went through a few steps of NM updates towards a self-defined loss function (NDCG@100 in our use case). NM-rankSVM converges within less than 10 steps of updates and outperforms many state-of-the-art learning to rank models by both performance and speed in offline global search experiements.

METRIC. Other than relevancy merics such as NDCG, a proper diversity metric could serve as a secondary offline measure to help us pick the right model for deployment. This allows us to evaluate the influence of record diversity on user engagements, thus measuring return on investment for different record collections. Furthermore, we need the metric to measure not only global diversification, but also local diversification. Global diversification measures how many record types are present in the list, while local diversification measures whether the same or different record types are present between row to row records. For example, if different record types are represented by letters, such as A, B, etc, and R_1^A represents that the first record is of type A, then the ranking list L_1 of $[R_1^A, R_2^A, R_3^B, R_4^B]$ has the same global diversity as an another list L_2 of $[R_1^A, R_2^B, R_3^A, R_4^B]$, but L_2 has better local diversification than L_1. Our assumption is that an optimal diversity list should cover as many record types as possible at any position.

To meet these requirements, we propose a new metric, which we term normalized cumulative entropy (NCE). NCE is inspired by how the ranking gain is accumulated and normalized in the definition of the ranking metric NDCG. It works in three steps. First, we use entropy to measure diversity in their search results, demonstrating its utility in quantifying global diversity. To further measure the local diversity, we propose to sum up entropy value at each position. Finally, we define an maximum entropy problem as an integer programming problem and use it to normalize the cumulative entropy in the range of 0 to 1. To the best of our knowledge, this problem has never been studied before. We propose a way to find its optimal solution in CLAIM 1 and skip the evaluation use branch and bound algorithm due to space limits.

Claim 1. *For K record types, the maximum entropy value of top n documents happens when there are (n/K) documents belonging to each of (K -n* $\mod K$*) record types. If (n* $\mod K$*) is 0, we are done. Otherwise for the (n* $\mod K$*) record types, there are (n/K)+1 documents belonging to each of these record types.*

For example, assume thre are 3 record types and we are interested in maximum entropy of top 5 documents, the maximum entropy occurs when the probability of each record types is $(2/5, 2/5, 1/5)$ according to Claim 1.

3 SPEAKER BIO

Peng Jiang is a senior research scientist at Ancestry working on search and recommendation with two patents filed. She has a Ph.D. in Computer Science at University of Illinois at Urbana-Champaign with speciality in numerical analysis.

Yingrui Yang is a data scientist at Ancestry. Currently she works on improving search relevance and decoding user search preferences. She holds a master in biostatistics from Harvard University with a research focus on causal inference and network analysis.

Alex Li is a senior manager of data science at Ancestry leading search and discovery. He has a Ph.D. in Computer Science at Michigan State University focusing on machine learning, information retrieval and algorithms.

Azadeh Moghtaderi is the head of data science at Ancestry, responsible for developing scientific solutions to problems within the product, business, and marketing domains. Azadeh holds a Ph.D. in Statistics from Queen's university in Canada. Her academic research was focused on problems of non-stationary time series analysis, in particular on the development of non-parametric techniques for time-frequency analysis, trend filtering, gap-filling, and prediction.