Query Understanding for Surfacing Under-served Music Content

Federico Tomasi, Rishabh Mehrotra, Aasish Pappu, Judith Bütepage, Brian Brost, Hugo Galvão and Mounia Lalmas
Spotify
{federicot,rishabhm,aasishp,judithb,brianbrost,hugog,mounial}@spotify.com

ABSTRACT

Platform ecosystems have witnessed an explosive growth by facilitating interactions between consumers and suppliers. Search systems powering such platforms play an important role in surfacing content in front of users. To maintain a healthy, sustainable platform, systems designers often need to explicitly consider exposing under-served content to users, content which might otherwise remain undiscovered. In this work, we consider the question when we might surface under-served content in search results, and investigate ways to provide exposure to certain content groups. We propose a framework to develop query understanding techniques to identify potential *non-focused* search queries on a music streaming platform, where users' information needs are non-specific enough to expose under-served content without severely impacting user satisfaction. We present insights from a search ranker deployed at scale and present results from live A/B test targeting a random sample of 72 million users and 593 million sessions, to compare performance of different methods considered to identify *non-focused* queries for surfacing under-served content.

CCS CONCEPTS

• **Information systems** → **Query intent**; *Query representation.*

KEYWORDS

query understanding; music recommendation; under-served content

ACM Reference Format:
Federico Tomasi, Rishabh Mehrotra, Aasish Pappu, Judith Bütepage, Brian Brost, Hugo Galvão and Mounia Lalmas. 2020. Query Understanding for Surfacing Under-served Music Content. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412741

1 INTRODUCTION

Modern platforms, such as Uber, Amazon, Airbnb, Spotify and YouTube, are increasingly emerging as the go-to applications facilitating economic exchange between consumers and suppliers. These platforms need to meet the needs not only of the demand side (*e.g.*, users), but also on the supply side (*e.g.*, retailer, artists). In the success of these platforms, search functionality plays a key role, as it links the direct needs of the users to relevant content provided by the suppliers.

Most traditional search systems have been user-centric in their approach of optimization and evaluation. Systems that power multi-stakeholder marketplaces however have to account for supplier exposure on their platforms. Typical of most search and recommender systems, there often exists content that is not surfaced to the users. This can be caused by a lack of metadata, popularity bias or unspecific query formulation. Further, blindly optimizing for consumer relevance has shown to have a detrimental impact on supplier exposure and fairness [22]. Indeed, an ill-optimized search system might unintentionally provide differential exposure to certain set of suppliers. To ensure a healthy, sustainable platform, systems designers need to consider this disparate impact on supplier exposure and develop ways of mitigating it.

Given the important role search plays in surfacing content, investigating ways to provide exposure to certain content and supplier groups via search results would go a long way in giving system designers more control over consumption on their platform.

In this work we consider the question: when might we surface under-served content in search results? Specifically, we consider a search system powering a large scale music streaming platform, in which artists pose as suppliers and users pose as customers. Recent findings around the non-specificity in user intents [19, 21] indicate that many opportunities to present under-served content exist on music streaming platforms. Users often have low or no preference about the specific content they wish to stream. Such opportunities can be exploited to surface under-served content, thereby enabling that content to reach a broader audience.

We propose a framework to develop query understanding techniques to identify potential *non-focused* search queries on a music streaming platform [19]. We posit that there exists certain *non-focused* queries with broad intents, where users are generally more open to non-specific recommendations (*e.g.*, "relaxing music"). Identifying such *non-focused* queries provides opportunities to surface result from undiscovered, under-served music content. This can help platforms to improve exposure of under-served content, without impacting user satisfaction.

To instantiate the problem and identify *non-focused* queries, we assume two types of content we wish to surface more, which we use as running examples throughout this work.

Niche Genres: Typical of most recommender systems, there exist categories of content not yet discovered by a broader audience. For example, genres like Classical, Jazz and Blues have historically been

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 ACM 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.
CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412741

considered *niche*, despite having dedicated groups of core listeners. For the less-familiar, non core-audience, search systems could surface this type of content more. This would make this content more accessible, and enable non-core listeners to discover it without alienating core-listeners.

Casual Music: Often users aspire to listen to casual music, mainly composed of non-music audio (*e.g.*, nature sounds) or instrumental tracks. Such music content is often characterized by smooth, cohesive composition and a lack of jarring or disruptive sounds. For this type of music content, users often have low or no preference on the specific content they wish to stream, and hence face trouble in articulating their information needs to access such content. This *casual* music forms the second group of content we wish to surface in search results, by identifying broad queries.

Our goal in this work is to identify queries for which a search ranker can potentially surface results from under-served content groups, such as the Niche Genres and Causal Music identified above, without hurting user satisfaction. Such a module can in turn be used by query-content matching and ranking systems to train models to surface under-served content.

Contributions. In this paper we consider the problem of identifying *non-focused queries*, *i.e.*, queries that can be targeted to boost under-served content. We draw inspiration from recent research on broad-intent queries and user mindset analysis to address the problem [19] and define *focused* and *non-focused* queries. Focused queries relate to a specific information need, *e.g.*, "*katy perry fireworks*". Non-focused queries represent a more broad and open ended information need, where users have a seed of an idea in mind and are generally more open to non-specific recommendations, *e.g.*, "*relaxing music*".

Building upon existing work on search query understanding [7, 13], we identify three classes of features derived from query characteristics and user interactions, which are predictive of identifying relevant queries to surface under-served content (Section 3). Furthermore, for each class of under-served content, we train models to predict a query level score that indicates the suitability to surface under-served content (Section 4). Due to the lack of large scale labeled data, we additionally present weak supervision approaches to learn from limited labeled data.

We present insights and findings from a large scale deployed search ranker powering a popular music streaming platform. Based on user interaction data from over 10 million users and 500K search queries, we analyze how the different query identification features perform, and present offline prediction performances across different models. Finally, we conduct a live AB test on 70 million users for a duration of one week to evaluate the predictive power of the proposed model in an online test. We contend that our findings and insights have implications on the design of multi-stakeholder search systems powering online marketplaces.

2 RELATED WORK

Search & Recommendations on Platforms. The major motivation of our work stems from the need to balance supplier preferences with user needs on multi-stakeholder platforms. Platforms and marketplaces have enjoyed a long history of detailed research [28, 29], with past work exploring competition [5], strategies [14] and economics [20] on such platforms. The concept of multiple stakeholders in recommender systems is also suggested in prior research [1]. The role of search and the need for supplier considerations is an understudied area, which we focus on in our work.

ery Understanding. To facilitate the exposure of under-served content, we rely on developing query understanding module. The problem of query understanding has enjoyed a long history of active research, with advancements around understanding query semantic [15], knowledge-based conceptualization [33], semi-supervised learning [32] and neural learning for voice query understanding on an entertainment platform [27], and query intent modeling [6, 16].

Sponsored Ads. An idea related to surfacing under-served content via search results is Sponsored Ads. Given a set of keywords, usually businesses pay for advertisements to show up in the search results when the users write a search query including those keywords. Previous works addressed this as multi-label learning problem where the words within a query are treated as labels to annotate the relevant results along with advertised content [2, 12, 26].

We build on top of such work and in particular on quantifying broadness of query aspects [30] and intent [6], and leverage such aspects to specifically consider how some queries are better suited than others to include under-served content.

3 QUANTIFYING NON-FOCUSED QUERIES

We want to detect *non-focused queries*, *i.e.*, candidate search queries for which the ranking algorithm could present under-served content to users while satisfying their search needs. An intuitive approach could be to check existing results associated with queries, and check for which queries the users are consuming under-served content. However, such content is by definition surfaced very rarely, hence the simple inspection of results for existing queries is largely ineffective. This presents a major challenge, as the direct inspection of queries and results shown and consumed by the user is non-informative for the majority of the queries. To overcome the limitation, we propose the following groups of features:

- **Standalone features:** these include surface-level information from the queries alone.
- **Reference dependent features:** these are conditioned on gold standard (reference) queries that already included under-served music content in their results, and consumed by users.
- **Interaction features:** quantifying the generality of a query.

Notation. We indicate with $\&$ all of the queries under analysis, and $\&^A$ indicates the reference queries, *i.e.*, those queries for which the users already are consuming under-served content. We refer to the results associated to queries as $'$. The results the users click on are indicated with $'^2$, while the results displayed (but not necessarily consumed) are indicated by $'^3$. The under-served content available on the platform (not necessarily included in any query result) is indicated as .

3.1 Standalone Features

There are few queries for which users are consuming under-served content. To augment this set of queries, the first features we compute is the number of under-served content displayed to users and the number of those consumed by the users for a particular query. Formally, the number of under-served content displayed for query q, indicated as $\#_{displayed}(q)$, is defined as $\#_{displayed}(q) = |\{r \in R_q : r \in U\}|$. Similarly, the number of under-served content consumed by the users for a query q, indicated as $\#_{consumed}(q)$, is defined as $\#_{consumed}(q) = |\{r \in C_q : r \in U\}|$.

In the current system, a query is served through Prefix Query Resolution mechanism, hence the text is often an abridged version of the actual user intent. Therefore, the query text itself has little context for feature extraction. We assume that the clicked results, e.g., track titles, album or artist names, are latent representations of the user's intent. We compute the embedding vector of a query q as a weighted average of the Word2Vec [23] vectors of the clicked results. Similarly, the vector representation of a track is computed based on its co-occurrence statistics across playlists and the vector representation of an artist is the weighted average of the tracks they have performed [4].

3.2 Reference Dependent Features

Very few queries are labeled as reference, meaning they include under-served content consumed by users. Nevertheless, results returned for these $q \in Q_A$ queries can be compared against results returned for all other queries $q \in Q$. We hypothesize that higher overlap among the pair of results would mean that a particular query q is highly similar to a query belonging to Q_A. We compute two similarities: overlap in clicked results (f_2), and overlap in displayed results (f_3). Both are computed in terms of the number of entities overlapping divided by the size of their union, i.e., the Jaccard similarity between consumed or displayed results. Formally, for any query q we compute $f_2(q) = \max_{q' \in Q_A} J(C_q, C_{q'})$, where J is the Jaccard similarity. Similarly, $f_3(q) = \max_{q' \in Q_A} J(R_q, R_{q'})$.

Embedding Distance. We conducted a pilot study and observed that, given the wide range of results available for the same query, the Jaccard similarity is 0 in most cases. To remedy this and avoid scalability problems when Q_A is large, we considering non-exact matching additional metric, the distance $d(q, \cdot)$, defined in terms of embeddings of the target entities' (artist, track, album, playlist, i.e., results that users click on for the particular query), with respect to under-served content. Formally, $d(q, \cdot) = \min(5, d(q, \cdot))$, where $\min(5, \cdot)$ computes the average of its 5 smallest inputs, and $d(q, q') = \|\vec{v}_q - \vec{v}_{q'}\|_2$. The distance is computed using FAISS, a fast approximate nearest neighbor algorithm [17]. The distance is useful when there is currently no under-served content displayed. Queries that exhibit a low $d(q, \cdot)$ while not including any of such content include study music, peace, and sleep stories.

Pronunciation distance (Dist Pron). This metric measures the weighted pronunciation distance between the reference queries $q' \in Q_A$ and rest of the queries. The pronunciation distance metric is a customized Levenshtein distance [18] that overlooks the orthographic differences to capture the similarity between the query texts. First we convert the query text into sequence of phonemes by applying Grapheme-to-Phoneme (G2P) model trained on a recurrent neural network (RNN) with long short-term memory units (LSTM) architecture [24]. Since queries tend to be very noisy, it is advisable to ignore commonly confused pairs of phonemes (e.g., {D, DH, T, TH}). To this end, we choose a lower edit cost for commonly confused pairs while computing the distance, otherwise all edit costs are 1 and the lower edit costs are derived from the statistics discussed in [3]. Then we compute the distance between pairs of q and q' phoneme sequences. An advantage of pronunciation distance over the lexical edit distance is the ability to detect common misspellings, word elongations, or incomplete strings of a reference query. The table below shows that the pronunciation distance is 0 for the pair of incorrect and correct query (Randy Rhoads pronounced as r ae n d iy r ow d z) which means they are highly similar, whereas the lexical distances are non-zero values.

Wrong spellings	Lexical distance	Phonetic spelling	pronunciation distance
Randy Roads	1	r ae n d iy r ow d z	0
Rhandy Rohads	3	r ae n d iy r ow d z	0
Rhaandhy Rhoadzzz	6	r ae n d iy r ow d z	0

Knowledge Graph Distance (Dist Wiki KG). Queries that often share similar results tend to share ontological roots. To capture this, we linked the queries to entities on Wikipedia Knowledge Graph (KG) using an open-source entity linking toolkit, Fast Entity Linker [8, 25], mapping the partial queries to canonical KG entities. Then, we measure the distance between the embeddings of KG entities (corresponding to their respective queries) using the pre-trained embeddings model Wikipedia2Vec [34]. Query text mapped to their respective KG entities could disambiguate queries that are lexically similar but refer to different entities and are distant in the embedding space. The example below shows queries with a common phrase small town but referring to different KG entities:

Query text	Wikipedia KG entity
small town usa	Small_Town_USA
small town girl	Small_Town_Girl_(song)
break up in a small town	Break_Up_in_a_Small_Town
small town saturday night	Small_Town_Saturday_Night_(song)

3.3 Interaction based Features

Click Entropy. This indicates whether a query is highly unfocused or not. For a query q the entropy $e(q)$ is computed as $e(q) = -\sum_{r \in R_q} p(r) \log p(r)$, where $p(r) = \frac{\#_{clicks}(r)}{\#_{clicks}(q)}$ is the probability of the result r to be consumed by the users. As this value is not analytically computable, we estimate it by counting the number of times that users clicked on a particular result based on the same query. Entropy indicates broad intent understanding. Simultaneously, there is a strong correlation between an unfocused query and receptiveness of a user to explore novel content [31]. Examples of queries that include under-served content and have high entropy are wedding instrumental, sad, morning.

4 PREDICTION OF NON-FOCUSED QUERIES

The proposed features carry information on which non-focused queries are better suited to return under-served content. Based on these, we want to learn a model to predict whether an unlabeled query could help presenting under-served content in the search

results. We devised two manual thresholding-based predictors, and three machine learning models using the proposed features.

4.1 Feature Thresholds as Predictors

First, we analyzed threshold-based predictors. For each feature under analysis, the predictor aim at predicting the output based on an evolving threshold. For threshold d_B and a feature B, we regarded the examples having the feature under examination higher than the threshold as positive, and those with the feature lower than the threshold as negative. Formally, the prediction $\sim_{CB} = 1$ if $B \geq d_B$, $\sim_{CB} = 0$ otherwise, where B is a non-distance-based feature. For distance-based features the prediction is reversed, because of the negative correlation of distance-based features with the output to predict. d_B is a feature specific threshold that has been cross-validated on a learning set $;_A$ after multiple splits in training and validation (10 fold cross-validation).

We also tested a pairwise combination between the features, based on a pair of features and thresholds using the logical AND between two features, as $\sim_{CB} = 1$ if $B_1 \geq d_1$ AND $B_2 \geq d_2$, $\sim_{CB} = 0$ otherwise. As before, when B_1 or B_2 are distance-based features, we consider $B_1 \leq d_1$ and/or $B_2 \leq d_2$.

We then employed an automated procedure to find the best multiple combination of features, using a decision tree classifier [11] to automatically devise a prediction rule to identify which queries could potentially include under-served content. A decision tree classifier builds a decision tree, meaning that starting from the most discriminative feature, it employs a set of threshold-based rules on the features to arrive at the best prediction. As this procedure is prone to over fitting [9], we employ a grid search cross-validation procedure limited to height tree structure of 5 to find the best decision tree. Not only do decision trees provide an automated way to infer the best thresholds, they also allow us to easily interpret the results through their inspection.

Each sample is associated with a probability of belonging to one class. Based on this, we consider two ways to assign the class label to the sample: we assign a class if the probability of the label is higher than 0.5, or we change the probability threshold after cross-validating the threshold on the validation set.

4.2 Trained Models

Threshold-based predictors have clear limitations, such as failing to learn non-linear relationships. To overcome this limitation, we use two statistical models trained on the features discussed above. First we trained a random forest classifier, an ensemble learning method that constructs a multitude of decision trees during training and returns the class that corresponds to the mode of the classes of the individual trees during prediction [10]. Due to random feature selection and bagging, random forests are known to alleviate over-fitting problems. We train and validate the model using a 10-fold cross validation procedure for optimal parameters. We also use a neural network architecture comprising two dense layers (with 128 and 64 neurons, respectively) with a dropout of 0.5 between the layers, which we train for 100 epochs.

4.2.1 Leveraging Weak Supervision.

Query labeling has a high cost. The majority of queries are unlabeled, hence we prefer to use the few labeled queries to improve the predictive power of our models. We employ weak supervision techniques with inaccurate labeling [35]. To assign a weak supervision label to unlabeled queries, we explored several approaches. First, we experimented on which model to use to assign the weak supervision label, between the decision tree and the random forest classifier, and we chose the latter as it performed best on the validation set. Then, we estimated the output labels of all queries $\&$ using the random forest classifier, and regarded such labels as weak supervision labels. Second, we considered different ways to use the weak supervision labels: (i) using only a subset of the weakly-labeled queries, based on the probability of label assignment, i.e., only considering the queries for which the model assigned a label with probability higher than 0.8, or (ii) considering all 500K queries with the weak supervision label, but using the probability of label assignment as sample weight during the training procedure of the neural network. We empirically selected the latter option, based on an higher performance on the validation set.

4.3 Ensemble

Each classifier, as empirically shown in Section 5, has its own strength and weakness. To benefit from all of the different classifiers, we devised an ensemble model based on the predictions of the best performing classifiers, namely the threshold-based (using Entropy, Pronunciation distance, Min Dist Content, Dist Reference Queries) and the neural network classifier. The ensemble model is a voting mechanism with three different voting criteria: $= {}^{1}@ = 0$ if at least one classifier identifies non-focused query, 1 if at least two classifiers agree, or 2 if at least three classifiers agree. Such voting criterion aims at higher recall: when any of the classifiers predict a query to be non-focused, then there is potential to present under-served content to the user.

5 EXPERIMENTS & RESULTS

We conduct three empirical evaluations to investigate the effectiveness of different techniques to identify queries to surface under-served music content. First, we gather labeled test data from music domain experts and evaluate the models on it. Then, we conduct gain vs. loss based comparison to showcase the trade-off between surfacing more under-served content and drop in relevance. Finally, we demonstrate performance gains via a live A/B test in Section 6.

Dataset Description. We use logged feedback data and live production traffic from an online music streaming service to understand how users search for music content. We logged the search result pages returned for each query along with users' interactions (such as clicks or taps) on the results. This dataset consists of 500K unique queries, extracted from a random sample of 35M queries in May 2020. The queries are selected if they include more than two characters. Also, we limit queries by only considering US users. Starting from the original 35M queries, those have been grouped together by the query text (exact matching after trimming and clustering of queries having at least the first three letters in common, based on the number of entities), which resulted in 500K queries.

Using a stratified sampling across the query population with entropy and distance features, we selected 1000 queries for manual annotation by a team of in-house subject matter experts in music content and culture. The annotation task was performed separately for the Niche Genre and Casual Music groups. For each query, an