Challenges in Search on Streaming Services: Netflix Case Study

Sudarshan Lamkhede*
slamkhede@netflix.com
Netflix Inc.
Los Gatos, California

Sudeep Das*
sdas@netflix.com
Netflix Inc.
Los Gatos, California

ABSTRACT

We discuss salient challenges of building a search experience for a streaming media service such as Netflix. We provide an overview of the role of recommendations within the search context to aid content discovery and support searches for unavailable (out-of-catalog) entities. We also stress the importance of keystroke-level Instant Search experience, and the technical challenges associated with implementing it across different devices and languages for a global audience.

CCS CONCEPTS

• **Information systems → Users and interactive retrieval**; **Recommender systems**; *Retrieval models and ranking*.

KEYWORDS

search, recommender system, user study, experimentation

ACM Reference Format:
Sudarshan Lamkhede and Sudeep Das. 2019. Challenges in Search on Streaming Services: Netflix Case Study . 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, 4 pages. https://doi.org/10.1145/3331184.3331440

1 INTRODUCTION

Streaming media services such as Spotify, iTunes and Pandora etc. for music, and YouTube, Netflix, Amazon Prime Instant Video, HULU, and HBO etc. for video have witnessed large scale adoption in recent years. Convenience, control and choice offered by these services have made them indispensable in broadband connected households [4].

Most streaming platforms provide users with access to vast repositories of content with only a small fraction familiar to them. Recommender Systems typically do the heavy lifting of providing the user with a relevant subset of items, e.g. the Home Page experience on Netflix, or the personalized Daily Mix recommendations on Spotify. However, Search on these services is critical for discovery and exploration of content. In particular, Search is the main avenue for new users to explore the content catalog, and for tenured users to break out of the so-called filter bubble [5]. On Netflix, too, the Recommendation System is the primary driver of discovery (act of watching content that was not watched at all by a user on our service, previously), and Search plays a complementary role to the personalized recommendations. We have observed that more than 20% of discovery streaming happens through Search on Netflix. Users have different use-cases for search on streaming media platforms that go beyond traditional information retrieval and require seamless integration of recommendations into search results. For instance, searches for videos that are not available for streaming ("unavailable entities") require relevant recommendations to be surfaced that are related to the unavailable entity. Also, unlike Web Search, a lot of search activity happens on disparate devices - smartphones, TVs, game consoles etc. posing unique problems of device adaptation. All these novel interactions lead to Search on streaming services being significantly different from Web Search or traditional IR systems.

2 SEARCH USE CASES: FETCH, FIND, AND EXPLORE

User studies at Netflix have revealed three different mindsets in which members interact with Search, namely, *Fetch*, *Find* and *Explore*, even though the aim is to watch something for entertainment. This is quite different than the intents of navigation, information, or transaction behind the Web Search queries [2].
Fetch: In this use-case, the users have a clear intent of retrieving a specific item from the catalog to stream. For example, a user who wants to watch Stranger Things may issue a query `stranger things`. In this case users want the system to immediately satisfy their needs by returning the entity. This is by far the most common use case and largely satisfied by traditional information retrieval techniques that rely on lexical matching.
Find: In this use-case, users have formulated their entertainment needs but they do not have a specific item in mind. An example search would be for `radhika apte`. The actress stars in multiple movies and TV shows and the user may be willing to watch any of those. The users' expectation is that the query is understood and relevant items are returned. They have the perception of a partnership between the them and the Search system that will enable them to narrow down the choice to a single video.
Explore: In this use-case, users typically enter much broader queries, such as `horror movies` or `spanish` and the idea is to explore the content in that general area. The role of the Search system is to provide a slate of relevant results and guide the user through a meaningful journey to a title they would finally choose to watch.

While discovery of content is the main goal of the explore use-case, it can happen in other use-cases too. Only the fetch use-case is supported by a simple lexical match. Therefore, Search needs to blend in both lexical and behavioral results to provide a richer, more meaningful experience.

[*] Both authors contributed equally to this research.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07.
https://doi.org/10.1145/3331184.3331440

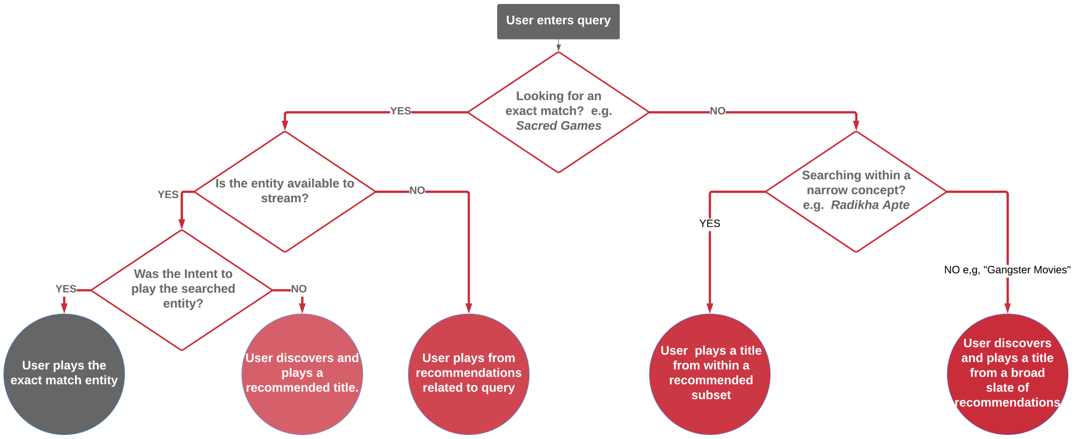
User enters query
Looking for an exact match? e.g. *Sacred Games*
YES
NO
Is the entity available to stream?
YES
NO
Was the Intent to play the searched entity?
YES
NO
User plays the exact match entity
User discovers and plays a recommended title.
User plays from recommendations related to query
Searching within a narrow concept? e.g. *Radikha Apte*
YES
NO e,g, "Gangster Movies"
User plays a title from within a recommended subset
User discovers and plays a title from a broad slate of recommendations

Figure 1: Possible paths connecting a query to a successful play event, illustrating the different use-cases of Search on Netflix. Non-lexical, behavioral recommendations become increasingly important for the terminal nodes shown from left to right. The leftmost terminal node is the only one that can be supported by purely lexical information retrieval.

3 ROLE OF RECOMMENDATIONS IN THE SEARCH CONTEXT

We define a **Search Result** as an entity retrieved by the search engine by matching query and context with the indexed entities. e.g. results from lexical matches. A **Recommendation**, on the other hand, is an entity selected by the search engine by relaxing the match constraints e.g. an entity retrieved via collaborative filtering.

The Search system has to retrieve the items that satisfy the query intent in the narrow sense (e.g. return all available James Bond movies if the query is `james bond`). Additionally, it has to delight users by helping them discover something that they would like to stream. Though the majority use-case (roughly 75% of searches) for searching is fetching videos by their titles, a significant portion of users engage in a more relaxed searching behavior. e.g. certain fraction of users would discover (and play) Piranha movies when they searched for `sharks` with a possible intent of watching Jaws. Some users like to co-search and co-watch videos for Bruce Lee and Jackie Chan. These intents are better served by recommendations.

In case of streaming music services, user studies have shown that serendipitous discovery is highly valued by listeners [6] and can increase their long term satisfaction with the service leading to continued subscription [7]. We hypothesize that those observations hold true for video streaming as well.

Providing meaningful recommendations within a Search experience poses two main technical problems:

(1) **What are good recommendations for a search?** - The user query and the search context put restrictions on what could be relevant recommendations. The query agnostic default personalized ranking that we show on the home screen, or unpersonalized, popular entities are not good recommendations in the search context. Such results could be too distracting and may not increase users' engagement or streaming activity.

So the recommendations need to be contextual and related to the intent. For example, Game of Thrones and Breaking Bad may not look similar based on the knowledge graph but both are highly binge-worthy shows[1]. So a user that is looking for the next binge-worthy show to watch, either would do. The relevance is also temporal. For example consider query `oscar nominees`. The intended result set is an ephemeral collection that groups disparate videos prior to the award announcements. We also want the recommendations to be personalized to some extent. When a user seeks related videos for The Mummy (1999) movie, they may tend to emphasize the "action-adventure" aspect over "horror", or "depicted-era" over "cast" in determining what is related. So we have to devise new ways of coming up with recommendations in presence of the extra information. Traditional query expansion techniques or pure co-play based result set augmentation are not adequate.

(2) **How to blend the recommendations with search results?** This can be cast as a re-ranking problem: we can get two sets of videos - search results and recommendations (as defined above) and have another ranking function to rank the combined set. Such blending introduces specific challenges:

(a) The final ranking should respect that strong lexical matches be surfaced at prominent positions when the intent is to retrieve that item e.g. we should be able to show a documentary Shark for query `shark` at a prominent position, and not bury it within behaviorally relevant entities. We expect Black Panther to be prominently placed in results when users query it by a prefix of the title.

(b) The final ranked result set should look relevant overall without any obvious quality issues or inadvertent sensitive or offensive entities that make users feel like the system let them down.

3.1 Unavailable Entities

The set of videos available to stream in a market (usually defined at a country level) is limited due to licensing requirements and business constraints. This problem of uneven video availability is described from Recommender Systems perspective in [9]. Even the tenured users are not aware of these restrictions. They often search for shows or movies that are not available for streaming.

[1] Shows that users may like to watch multiple episodes of, in rapid succession

We estimate that at least 13% of searches on Netflix are for out-of-catalog videos. We need to detect their desired intent and entity, and if the entity is not available for the users to stream, we have to provide them with relevant and delightful recommendations that the users are likely to stream in absence of the original entity.

This problem arises in E-Commerce setting as well [11, 12] (e.g. a retailer may not carry a particular brand but not other) but is not noticeable for Web Search backed by a comprehensive crawled corpus spanning the entire Web.

We do not want the users to hit a dead-end in their efforts to find something worthy of streaming if they issue an unavailable entity search. This goes beyond just plainly recommending related or similar entities. And personalization can play a key role here. There are three distinct sub-problems here:
(1) *How to detect that the query is for an unavailable entity?* We could index all known entities in the domain of movies and TV shows and then match the user query against those. Keeping such knowledge base up to date and accurate in all languages is very resource consuming. Further, the query could match available and unavailable entities simultaneously and/or match multiple unavailable entities. In such cases, narrowing the intent becomes harder.
(2) *Whether to and how to message the user about unavailability?* Even when we unambiguously know which entity user queried for, messaging it back to the user is nontrivial from the UI perspective. Additionally, it is unclear what effect the messaging would have on the users' perception of the service.
(3) *How to provide substitutable entities* i.e. entities that can also fulfill the broader intent but aren't exactly the requested entity? Unavailable entities are not present in the co-play data so the traditional collaborative filtering models cannot derive item-item similarity for them. Pure metadata based similarity leads to sub-optimal user experience as it fails to account for user behavior patterns. Also, depending on the searched entity, it is possible that users would not stick to their original intent if the entity unavailable.

4 INSTANT SEARCH

For long-form video entertainment, users are presumably in a laid-back consumption mode. With least amount of interaction and cognitive effort, they would like to satisfy their entertainment need. Also, most of the viewing happens on TVs. Unlike handheld devices, their on-screen keyboards (OSK) are hard to use with remote controls or pointers. To type a single character, multiple movements of the cursor maybe required on the TV OSK. Voice search isn't that ubiquitous yet and second-screen experiences are not seamless. So to reduce the need to enter multiple characters, the Netflix service offers "Instant Search" meaning with every keystroke we provide a set of useful results, instantly.

Instant Search was found to have higher success rate and lower time-to-find in a separate analysis of query logs [3]. If users are not sure about their information need, e.g., if they do not know the correct spelling of the name they are searching for, they will probably make mistakes during typing. Similar to auto complete or query suggestions, Instant Search can guide users along the typing process allowing them to notice and correct mistakes as quickly as possible so that they need to enter as few keystrokes as possible to get to desired result(s).

Table 1: Query Length on Different Devices: Tokens (Characters). Instant Search leads to very short queries

Platforms / UIs	1%	25%	50%	75%	99%
Android OS	1 (1)	1 (4)	1 (5)	2 (8)	5 (23)
iOS	1 (1)	1 (4)	1 (5)	2 (8)	4 (20)
TV UI	1 (1)	1 (2)	1 (3)	1 (5)	3 (15)
Web UI	1 (2)	1 (4)	1 (6)	2 (9)	4 (21)

While instant search is very helpful for users, it introduces some technical problems. First, it makes the queries very short, second, it makes latency requirements stricter and third, different metrics and indexing schemes are required to optimize the experience.

4.1 Short Queries

The median number of tokens in query in Netflix's query logs is just 1 (table 1). In comparison, the average length of typed queries on Web Search was found to be 2.35 terms [10]. The queries on Netflix Search are short in number of characters they contain, too. They are even shorter on the TVs - median length is just 3 characters! This makes it difficult to use traditional approaches for query understanding and rewriting. One such example is spell correction. Due to partial queries, it is harder to detect whether the query is misspelled or just incomplete and offer appropriate corrections automatically. The differences in median query length on different platforms / UIs are due to the relative ease of typing. TV UI is the hardest so queries are shorter while Web UI on computers is the easiest so users type longer queries on it.

4.2 Latency

Users are more likely to perform clicks on the result page that is served with lower latency according to the large scale query log analysis described in [1]. This is probably true for all search services. Further, Instant Search results need to be rendered almost as soon as user enters a keystrokes. If the results are not instantaneous, user experience degrades, lowering user satisfaction. This requirement puts even stricter latency constraints - both on the UI and the back end - compared to other forms of search. Under these constraints, it becomes crucial to save every millisecond of computation which severely limits design of query understanding, retrieval and ranking components. We have to perform the computation, as much as possible, apriori and rely on efficient and clever caching schemes at serving time. The UI has to render the slate of result without any flickering or jarring updates to the view. Since UI is making multiple requests for a single search, there is more potential for timeouts and errors especially over unreliable network connections.

4.3 Metrics and Indexing

One of the objective of offering Instant Search is to help users find what they want to watch with least number of interactions. So driving down number of *keystrokes* to get a desired video in the view port is important, even though it leads to shorter queries that are harder to understand. This is different than traditional search ranking metrics which do not take into account number of interaction required to enter the query. Note that higher number of keystrokes may be required to type a query than the number of characters it has. This can happen due to query reformulation or the language's input method may require it. For example, Korean is usually typed using the Hangul alphabet where syllables are composed from individual characters. For example, to search for 올드보이 (Oldboy), in the worst possible case, a member would have to enter nine characters: ㅇ ㅗ ㄹ ㄷ ㅡ ㅂ ㅗ ㅇ ㅣ. Using a basic indexing for the video title, in the best case a member would still need to type three characters: ㅇ ㅗ ㄹ, which would be collapsed in the first syllable of that title: 올. In a Hangul-specific indexing, a member would need to write as little as one character: ㅇ.

5 GLOBAL AUDIENCE AND CONTENT

Many streaming services are global e.g. Spotify is available in 78 countries. Netflix is global - roughly 60% of Netflix's members are outside US and and a significant minority do not consume content in English at all. Netflix is localized in 22 languages and that list is growing. Though English is the top language for users on Netflix Search, less than 59% of users use it for searching (figure 2). This proportion is likely to go down as we localize our service in other languages and continue to grow internationally.

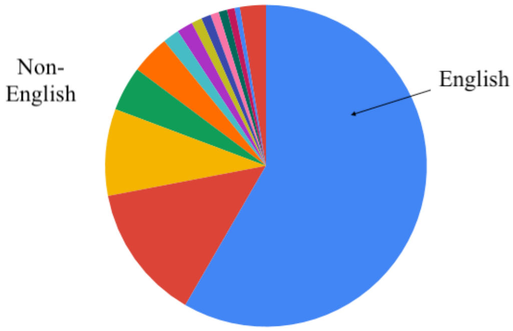

Figure 2: More than 40% of users search in a language other than English

We have to offer Search that operates well across all languages, countries and regions. In that sense, the challenges are similar to those in Web Search which also needs to tune the experience in each market. However, for Recommender Systems and Search on Netflix the differences in country specific corpora and local languages are more fundamental. Each market greatly varies in inclination towards local vs. international content, cultural tastes, query patterns, and content availability. Nonetheless, we are looking into transfer learning from US English to other locales.

How Netflix on boards a new language in Search is well described in the blog post [8]. Though our Search works on semi-structured documents that are relatively cleaner and smaller compared to the crawled documents from the Web or e-Commerce product feeds, the localization of the content in all supported languages is usually a challenge. A large number of movies and TV series are released every week all over the world. Keeping the knowledge base updated with all the released and yet-to-be-released entities across the globe with appropriate localization of text while maintaining the knowledge base integrity and high quality is expensive.

Localization sometimes poses an interesting challenge. While localized titles are easy to understand in the target market/locale, their original title may become so popular that users search them by the original title (e.g. La Casa De Papel was localized as *Money Heist* in English) causing a mismatch between the indexed string, title image and user query.

6 DISCUSSION

This paper describes how the unique user expectations from Search on a streaming media platform warrant approaches that need to go beyond traditional information retrieval and lean more toward behavioral data. Search and Recommender Systems need to work hand-in-hand to increase user joy via content discovery. Many of the challenges stem from the users' intent to play an entity that may be unavailable to stream, or their desire to explore the catalog via Search, the limitations of input devices prompting shorter queries, as well as the multi-lingual aspect of search for a global audience. While the specific task, user interface, and user interaction mode may differ between services, we believe that these challenges are relevant for Search on all streaming platforms. We hope that the novelty and practical importance of these problems will attract researchers, both in industry as well as in academia.

ACKNOWLEDGMENTS

We are thankful to Jon Sanders, Aish Fenton, Yves Raimond, and Justin Basilico, for their helpful feedback. We are grateful to Priya Kothari for the invaluable user studies narrative and to Yves Raimond for the example of keystrokes matching in Hangul locale.

REFERENCES

[1] Ioannis Arapakis, Xiao Bai, and B. Barla Cambazoglu. 2014. Impact of Response Latency on User Behavior in Web Search. In *Proceedings of the 37th International ACM SIGIR Conference on Research Development in Information Retrieval (SIGIR '14)*.

[2] Andrei Broder. 2002. A Taxonomy of Web Search. *SIGIR Forum* 36, 2 (Sept. 2002).

[3] Inci Cetindil, Jamshid Esmaelnezhad, Chen Li, and David Newman. 2012. Analysis of Instant Search Query Logs. In *Proceedings of the 15th International Workshop on the Web and Databases 2012, WebDB 2012, Scottsdale, AZ, USA, May 20, 2012.* 7–12.

[4] Leichtman Research Group Inc. 2017. 64% OF U.S. HOUSEHOLDS HAVE AN SVOD SERVICE. https://www.leichtmanresearch.com/wp-content/uploads/2017/07/LRG-Press-Release-2017-07-24.pdf

[5] Ray Jiang, Silvia Chiappa, Tor Lattimore, Andras Agyorgy, and Pushmeet Kohli. 2019. Degenerate Feedback Loops in Recommender Systems. arXiv:arXiv:1902.10730

[6] Jin Ha Lee, Hyerim Cho, and Yea-Seul Kim. 2016. Users' Music Information Needs and Behaviors: Design Implications for Music Information Retrieval Systems. *J. Assoc. Inf. Sci. Technol.* 67, 6 (June 2016).

[7] Matti Mäntymäki and A.K.M. Najmul Islam. 2015. Gratifications from using freemium music streaming services: Differences between basic and premium users *(ICIS)*.

[8] Ivan Provalov. 2016. Global Languages Support at Netflix. https://medium.com/netflix-techblog/global-languages-support-at-netflix-testing-search-queries-ede40f7d93d3

[9] Yves Raimond and Justin Basilico. 2016. Recommending For The World. https://medium.com/netflix-techblog/recommending-for-the-world-8da8cbcf051b

[10] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. 1999. Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum* 33, 1 (Sept. 1999).

[11] Gyanit Singh, Nish Parikh, and Neel Sundaresan. 2012. Rewriting Null e-Commerce Queries to Recommend Products. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12 Companion)*.

[12] Gyanit Singh, Nish Parikh, and Neel Sundaresn. 2011. User Behavior in Zero-recall Ecommerce Queries. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*.