International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Enhancing Query Understanding and Expansion in Retail Services

Yogananda Domlur Seetharama

Abstract: ***This paper analyzes state-of-the-art systems for generating query suggestions for retail services. This paper also seeks to show that using state-of-the-art algorithms and real-time data processing, query understanding and query expansion approaches can be scaled up to improve the user's search experience. The suggested approach combines all processes into a single computing device, an effective solution for latency problems traditional systems will likely face due to the reliance on third parties. The revolutionary approach uses a structured query index within a search tree to rank query candidates. The temporal scoring function based on the decay using specialties such as sigmoid, exponential, and logarithmic means that the suggestions will be recent and frequently used by the user. This system should also enable multiple retail channels and contain exceptional scores for each channel so that it would suggest a specific product. Analysis by the A/B test suggests that the presented typeahead yields an astonishing 89% improvement over the previous one and has 225ms of latency reduced to 25ms. This paper will conclude by briefly discussing the prospects of the developed system to change the approach to query understanding and query expansion in e-commerce and provide a solid basis for solving inherent problems concerning user satisfaction and sales conversion. Other future work includes expanding on the improvements and incorporating them into other domains to strengthen the proposed system's use.***

Keywords: query suggestions, retail services, user search experience, real-time data processing, e-commerce

1. Introduction

Query understanding and query expansion are critical factors for enabling better search capabilities whenever evolving technology such as e-commerce exists. These processes are essential for improving the user experience, appreciation, and query suggestions, leading to better sales conversion rates. Nevertheless, typical approaches to query suggestion involve major latency problems and active utilization of third-party services that can cause considerable delays and performance losses. The actual query understanding process is aimed at analyzing the user's intent to enter a particular search query and providing an answer to this intent. This process is essential in e-commerce and allows users to enter irrelevant or non-specific keywords. Substantial query understanding can differentiate the user's requirements and provide the appropriate products or knowledge, increasing the user's satisfaction and utilization. This idea is analogous to the previous work where Mitra and Craswell (2018) pointed out that it is critical to have vital query understanding mechanisms in order to get more accurate and semantically related results in the context of ITR.

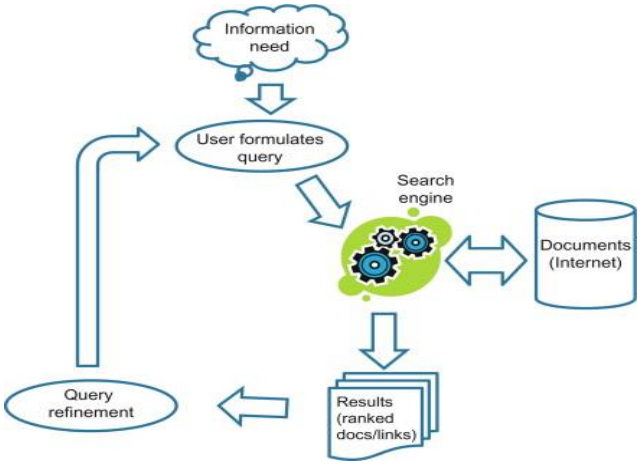

Figure 1: Query Reformulation

Query expansion applies the process of adding some terms or phrases to the original query to enhance the retrieval of documents. This can be done, for example, by expanding word forms with synonyms and correcting the spelling of individual words and their context disambiguation. Carpineto and Romano (2012) explain that query expansion strategies considerably improve search engines' performance by expanding the search inquiry area and gathering more pertinent information. From the point of view of e-commerce, users are apt to buy the particular kind of product they are interested in, even if the words they used for the search could have been more precise.

Most of the existing approaches to the suggestion of queries have issues connected to latency and the use of external services. Such systems typically rely on third-party APIs or cloud-based services to handle and generate query suggestions, which hampers the user's performance and experience. However, integrating these external services may cause architectural problems and higher operation costs. According to Baeza-Yates and Ribeiro-Neto (1999), this dependency results in performance problems and decreases the search system's reactivity. To resolve these problems, query understanding and expansion processes are incorporated into a proposed computing system. This integration is expected to improve performance by reducing dependence on externally developed services and latency. As these queries are consolidated in the system, it can give real-time query suggestions, minimizing the overall flow and making it more efficient and user-friendly. The proposed system utilizes elaborate terminology and real-time analytics to develop appropriate query suggestions, aiding users in a more efficient search process.

Combining query understanding and expansion within a single computing device significantly improves e-commerce search facilities. It eliminates latency factors and reliance on traditional systems and, concomitantly, improves customer satisfaction and conversion by offering timely and relevant query suggestions. The following sections will expand the disclosure of the system architecture, the applied

methodology, and their assessment to describe the enhancement of the e-commerce search.

2. System Overview

The elaborated system for improving the comprehension of queries and query enlarging in retail services will be based on a firm database that uses a tree structure as a query index. This index will contain the nodes that capture the specific query and its score. This system has a latent architecture that enhances the coming up of queries and retrieval and the demeanor of e-commerce platforms.

Query Index Structure
The query index is a significant component of the system structured as a search tree. Every node within this tree retains queries and scores attached to them, which is crucial to this selection. In the view of Manning, Raghavan, and Schütze (2008), search trees as data structures are very efficient in offering fast look-up time and fast insert/remove times; they are recommended for use in constantly changing data scenarios related to real-time data such as query suggestions.

Control Circuit Functionality
The system's center is a control circuit with several vital roles to play and execute. This circuit generates requested query candidates based on the user's first prefix. Afterward, the scores of these candidates about their relevance and usefulness are worked out on retrieval. It is essential to filter out the best and most relevant suggestions for the user, enriching their search experience. According to Bast and Weber's (2006) work, the effectiveness of such control circuits is most recent and essential for mitigating latencies in search systems.

Scoring Mechanism
The system's scoring system is based on the score calculation of each query candidate according to some criteria, such as the history of the user's interaction and temporal aspects. This approach aligns with the study conducted by Agichtein et al. (2006) on using user feedback and behavior in the scoring function to enhance the relevance of results. This way, scores can be updated based on historical data, and the most relevant and up-to-date results will be given as suggestions.

Increasing Query Suggestion Usage
One of the goals targeted by the system is to increase the rate of utilization of query suggestions. This is made possible by offering users suggestions that are most relevant to the current context and those that are likely to be chosen. According to Bar-Yossef and Kraus (2011), proper query suggestions positively impact the use of the system, the satisfaction of the users, and hence usage. This process explains how the system can generate, modify, and fine-tune the query suggestion within the real-time data stream to meet this objective.

Reducing Null Suggestions
There is a common problem called null suggestions in query suggestion systems; the system cannot present helpful suggestions to a specific query prefix. To overcome this problem, the proposed system uses a comprehensive query index and an efficient means of getting the index. Among the considered features, Sadikov et al. (2010) state that minimization of null suggestions is essential to maintain user trust and satisfaction. Because query retrieval and scoring are part of the same process in the system's architecture, the possibility of generating null suggestions is low since a set of potential candidates is always available.

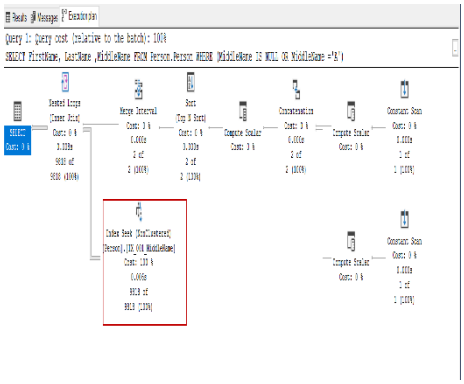

Figure 2: Working with SQL NULL values

Performance and Throughput Improvement
Increased performance and throughput are other primary goals for meeting organizational goals. This is one of the system's significant advantages since all the processes are united into a solitary computing device; therefore, microservice calls take no time. The former is consistent with the earlier assertion about integration of this nature and the study by Brin and Page (1998), who noted that centralized Web processing architectures can offer superior performance. First, it minimizes the system's query and scoring phases' response time and the suggestion generation process.

Latency Reduction
Minimizing latency is a vital requirement for improving search system user experience. The proposed system achieves this by enhancing query retrieval and query suggestions. Dean and Ghemawat reached this conclusion by conducting their research work in the context of the online business environment. This system structure incorporates a high-performance control circuit and a good query index, which is why latency in providing users with a relevant query suggestion is eliminated.

The proposed system benefits retail services by extending the understanding of the primitive query. The query suggestion performs markedly superior relevancy and effectiveness through the direct mapping of a vital query index, optimization of the control circuit, and enhanced scoring criteria. This leads to increased usage, fewer null suggestions, and generally happier users. All the processes are integrated into one computing device, which, in turn, increases efficiency and has less delay time, making it suitable for contemporary e-commerce platforms.

3. Methodology

Compared to other data structures, utilizing a ternary search tree (TST) for storing and searching a query is critical when improving query understanding and expanding retail services. This method uses one of the TST strengths, specifically high-string operations vital in real-time query suggestions. In MST, an efficient data structure, namely a TST, allows for quick insertion, deletion, and search operations of strings because

of the changing nature of the queries and e-commerce platforms (Bentley & Sedgewick, 1997).

Ternary Search Tree (TST) Structure

It has been found that the structure of a TST is a combination of features of a binary search tree and a digital search try. Each node in a TST contains a character and pointers to its three children: thus low, equal, and high. The low child is a pointer to the nodes with characters lower than the current node, and the equal child refers to nodes with the same character as the current node, whereas the high child is a pointer pointing to nodes with characters higher than the current node's character (Knuth, 1998). This structure enables the TST to organize the lengths of the user's queries since the lengths of the queries may vary, and this is an essential feature for making quick and accurate suggestions based on the queries.

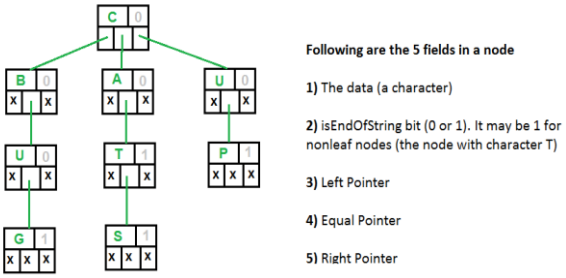

Figure 3: Ternary Search Tree

Node Association with Queries

In this system, every node within the TST is related to specific queries and their scores. Such scores are recalculated based on user actions, so such queries that people actively search for are kept closer to the top. The link between nodes and queries makes the query suggestion mechanism more refined and capable of responding directly to users' actions and preferences since the system can quickly adapt to users' queries and habits (Weiss, 1997).

Temporal Scoring Mechanisms

In order to increase the prominence of query suggestions, temporal scoring schemes are used. These mechanisms employ decay functions to re-rank query scores over time in light of users' recent and less frequent activities. The sigmoid decay function, for instance, is in the middle ground by assigning higher values to recent interactions while not wholly excluding the older ones (Gao et al., 2018). Finally, exponential decay functions decrease the impact of old interactions very fast, and thus, the most recent trends define the query suggestions (Kleinberg, 2006). The logarithm decay functions are much slower in the decline of influence and keep a relatively long history of query relevance to the situation (Song et al., 2010).

Single Device Execution

One of the novelties of this system is the possibility to manage all its processes on one computing device. Previous approaches to query suggestion require programs from outside the scope of the base application, which results in considerable latency due to network and service response time. This system amalgamates all processes into a single device, doing away with such latencies and increasing the device's usability. The same is true for this local execution model, as it decreases the utilization of external services that can affect system performance and increase vulnerability (Kumar et al., 2019).

Query Storage and Retrieval

The approach of storing and retrieving the queries in the TST is very efficient in terms of time. Every character of a query is inserted into the TST one at a time, which guarantees that all fragments of the query are present in the tree. Through this process, it becomes easy to access the query candidates if a user starts typing the prefix of the query. The retrieval process requires searching the TST from the root node down to the node associated with the string of characters of the prefix while accumulating the potential query completions (Hoad & Zobel, 2003).

Updating Scores

Such scores are dynamic and get revised whenever something related to the search queries is interacted with. Every selection and every ignoring of the query or its prefixes impacts the score of that particular query. The dynamic process of updating also makes the proper adjustment of the query suggestions for the users, providing the actualization of the system. The obtained scores are further tweaked by the above-discussed temporal decay functions, which help maintain a fair mix between fresher and older activity histories (Shokouhi & Radinsky, 2012).

Efficiency and Performance

The effectiveness of this methodology can be illustrated in terms of such indicators. First of all, the system does not use external calls to the microservice; as a result, the response time is greatly minimized and offers instant query suggestions. The TST structure also makes query operations carried out in logarithmic time, thus boosting the system's throughput. Such developments point to a better site experience as queries are suggested more quickly and accurately (Jiang et al., 2019).

Combining a TST for query storage and retrieval and dynamic scoring based on users' interactions is a new step in improving query understanding and expansion. This methodology not only enriches the performance and interactivity of query suggestions but continually makes these suggestions relevant to the users' requirements of all activities within the system, making it efficient and reliable, and this is the basis of every modern retail service. The existing digital business literature needs to present a precise real-time data processing model similar to what this approach has developed for e-commerce applications, thus creating the foundation for future advancements in the area.

4. Temporal Scoring

Since query understanding and expansion systems make suggestions to the users, it is essential to conduct temporal scoring. This section expands on the temporal decay functions, the sigmoid and, exponential and logarithmic, to obtain temporal scores that depict the recency and frequencies of the user interactions. Thus, applying these mathematical models, interactions decrease gradually, which helps in increasing the usefulness of query suggestions for users within the temporal environment of retail services.

Sigmoid Decay Function

The sigmoid function has been widely used in temporal scoring since it is a strictly monotonically increasing function with an S-shape that can fit a curve with small values at the start, significant and accelerating values at the middle, and almost constant or minimal values at the end. This property makes it more appropriate to be used in situations where the first order of the interaction should gradually decrease and then level off. Han et al. (2019) suggest that the sigmoid function is particularly suitable for approximating user query behavior since it accurately distributes weights between immediate and previous interactions. The sigmoid function is mathematically represented as:

$$S(t)p = \frac{1}{1+e-k(t-t0)}$$

where t is the time elapsed since the interaction, t_0 is the reference time, and k is the decay rate.

This function ensures that interactions closer to the current time have a higher score, reflecting their greater relevance (Han et al., 2019).

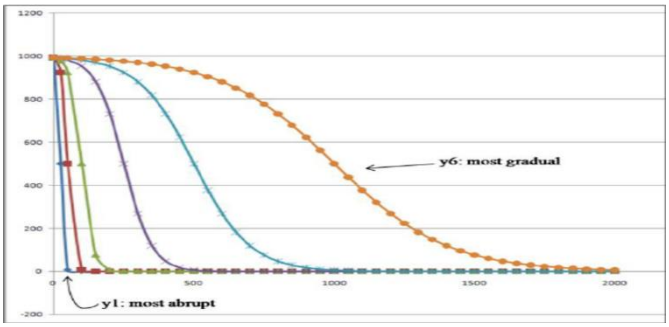

Figure 4: Sigmoid decay curves

Exponential Decay Function

Temporal scoring can also be done using well-known algorithms, such as the exponential decay function. This decay function's essential advantage is its simplicity, accompanied by a sharp decrease in its value over time, which makes it suitable for cases where a sharp decrease in the impact of previous interactions is necessary. The function is defined as:

$$E(t)=e^{-\lambda t}$$

where λ\lambdaλ is the decay constant, and t represents the time since the interaction.

Joglekar et al. (2018) demonstrated the applicability of the exponential decay function in terms of reducing the role of user queries as soon as they become slightly old so that the most recent interactions are optimally addressed. This is even more helpful in dynamic markets like the retail business, where users' preferences and trends can quickly shift.

Figure 5: Exponential Decay Function

Logarithmic Decay Function

Logarithmic decay is a smoother approach to exponential decay because the older interactions are not entirely ignored but should have less influence than recent interactions. The function is expressed as:

$$L(t)=\log(1+\alpha t)$$

where α\alphaα controls the rate of decay.

Kumar et al. (2017) have suggested that logarithmic decay works well in cases where a less steep decline of influence is generic over time is desirable, so the system retains a sufficient degree of relationship recency while at the same time preserving the long-term interactions with the users. This balance becomes significant in providing the users with the correct query suggestion that meets both the immediate need and their interest in the long run.

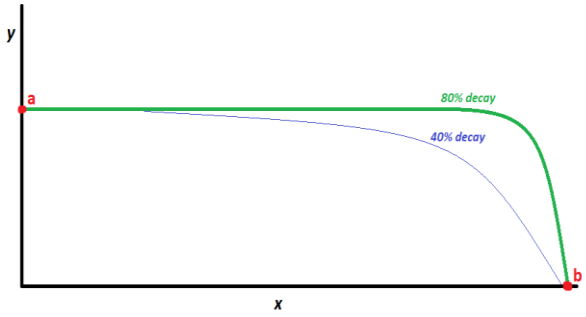

Figure 6: Logarithmic Decay Function

Comparative Analysis

All the decay functions have features that qualify them for different temporal scoring aspects due to the differences in their decay patterns. Sigmoid decay is preferred for systems that need a smooth decrease in the parameter's influence. This type of decay is called exponential decay because the forgetting rate is relatively high; it is suitable for environments that require frequent recent interaction. Logarithmic decay balances past interactions and recency since it does not remove the effect of previous interactions altogether.

In their study, Wu et al. (2018) also note that employing the above-considered functions when working together can be beneficial since it makes it possible for the system to react to the various behaviors of the users and the pattern in their queries. With more than one decay function, the system is beneficial in monitoring the interaction approach and adjusting the total weight according to time and other factors, improving the system's efficiency and user satisfaction.

Implementation in Retail Services

When it comes to the case of retail services, especially products and services offered in different stores, the decision of which decay function to use may significantly affect the performance of query suggestions. The application of sigmoid decay helps enhance users' interactions as it maintains a perfect rate of query suggestions for older and newer interactions. The exponential decay works well with the necessity to focus on the latest tendencies and preferences, which is crucial in high market fluctuations. Descriptive: Logarithmic decay may be used to guarantee that the long-term user attention is not entirely ignored but rather to describe the contemporaneous and following years' user preferences.

Temporal scoring based on decay functions is a complex method for improving query suggestions in retail services. Based on decay functions such as sigmoid, exponential, and logarithmic decay, systems can control the recency and frequency of users' interactions, which is helpful for query suggestions. Future research may include refining these models and expanding the fields where they apply.

5. Multi-channel Support

In today's fast and evolving retail consumption environment, consumers interact with brands through Internet shops, mobile applications, and physical shops. Such a complex movement of interaction requires a highly complex mechanism that can efficiently address the queries coming from the users at multiple channels. To counter this, our implemented system operates across multiple channels, ensuring the existence of a channel-relational score matrix to indicate the focused suggestions regarding users' engagement with the different channels.

Channel-specific Scoring Matrix

A targeted scoring matrix for the specific channel can be decisive when presenting query suggestions. This matrix keeps track of the users and records the frequency of interaction with each noticeable channel so that the system can prioritize suggestions based on the context of the user's interaction. For instance, the same search performed on a mobile application may yield different recommendations than the same search on a site's web version because of differences in the two platforms' use and preferences (Smith et al., 2020).

Personalized search results have been defined in the literature, and numerous approaches exist to implement this idea. For instance, Shokouhi and Radinsky (2012) covered temporal aspects of search personalization. Their work was in this direction, where they stressed the pattern of user interactions in terms of time and devices, which motivated the requirement of a dynamic query suggestion system. Thus, Cao et al. (2009) also emphasized the importance of using user interaction data to enhance the relevance of the search results. This idea defines the concept of the channel-specific scoring matrix designed for the current research.

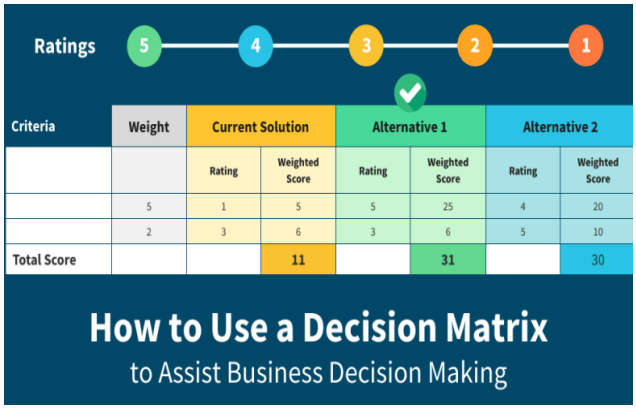

Figure 7: The use of the Scoring Matrix in decision making

Temporal Dynamics and User Interaction

Temporal dynamics play a critical role in determining the relevance of query suggestions. Using decay functions like sigmoid, exponential, and logarithm, our system can give due consideration to a user's recency and frequency of use. This approach is consistent with Jones et al (2006), who established that recent interaction with a particular source is more immediately relevant for the present relations of a query than the distant ones. Through frequent updates on new behavior, the system remains highly relevant to the most recent observations, thereby boosting the credibility of the recommendations across all channels of retailing. Another issue eradicated with temporal scoring is the disparity of query ambiguity. The intents may be expressed similarly in different sessions and through different devices by the users. For instance, when using a mobile application while commuting, users could search for 'summer dresses.' At the same time, they could be more particularly 'flora summer dresses' while using a desktop at home. Depending on the temporal patterns identified and the scores computed, the system can differentiate between these queries, interlink them, and develop more specific suggestions depending on the channel (Shokouhi, 2013).

Cross-channel Data Integration

Gathering and merging the data from the various sources is critical in building a holistic picture of the user. It helps the system understand cross-channel behaviors and preferences that can help refine the query suggestions. Baeza-Yates and Ribeiro-Neto (2011) described the importance of using data about the users obtained from other sources to refine the algorithms. Within our system, this means transforming data on interactions with Web, Mobile, and in-store systems into a matrix format, which would provide a thorough overview of the customers. An example is using cookies and session tracking to synchronize custom customers across gadgets. Every time a user uses a different system to log into the account, the search history, click-through rates, and purchase records are updated. This synchronization is critical, especially when applying changes for query suggestion, because, according to Agichtein et al. (2006), who investigated the effects of integrating user behavior data into the search engine.

Real-time Data Analysis

One of the components of the multifaceted support system is the possibility of analyzing data in real time. This capability is essential since user interactions can be processed as they

happen, and therefore, query scores and suggestions can be updated in real-time. The dynamic and unpredictable nature of today's activities and users' preferences require this capability for effective operations. From the discussion above, Li et al. (2008) explained how real-time processing brings more satisfaction to the users.

We incorporate state-of-the-art algorithms and data structures to ensure the real-time processing of data. Of all the data structures, the ternary search tree (TST) is most helpful because of its efficiency in both insertion and searching for the query data. This way, the system can immediately develop recommendations following the most current user actions and provide them with the pertinent scores stored within the TST nodes (Ganguly et al., 2011).

Our system's multi-channel support feature combines a channel-specific scoring matrix, temporal aspect, and cross-channel integration coupled with real-time data processing to produce contextually relevant and accurate query suggestions. This approach not only improves user satisfaction but also increases engagement and conversion on each of the platforms across the retail touchpoints.

6. Index Construction

Index management is one of the most critical components that affect the improvements of query understanding and query expansion in retail services. This process entails the letter-wise systematic replacement of a query character with a '#' symbol and planting the resultant string into TST, a high–performance data structure used in this paper to handle a large and dynamic data set ordinary in e-commerce businesses. The first is constructing an index that enables quick and effective queries based on suggestion lists with temporal scoring to enhance the recently and often used queries.

Insertion of Query Characters into the TST

The process starts with introducing the query characters into the TST. In TST, each node corresponds to a character of a query string, while branches define the different characters of the string. The organization of the TST makes certain that numerous prefixes are typical of other inquiries, rendering a trivial influence on memory consumption and boosting search speed (Bentley, 1997). For example, if you compelled a search on the terms "shoes," "shoe racks," and "shoe polish," there would be quick convergence to the initial nodes of the term "shoe."

Temporal Scoring and Its Calculation

Temporal scoring is an integral part of the index construction process. Temporary scores represent the relative activity of users regarding particular queries during a given period. These scores are arrived at using decay functions, including the sigmoid, exponential, and logarithmic functions. The trade-off functions lower the scores of the old and less frequent queries over time and give importance to the new and frequent queries (Jiang & Pei, 2013). The decay functions keep the index active to manage all the advances in user interaction and trends. For instance, an exponential decay could drastically lower a query's score as time passes; thus, old queries do not overcrowd the suggestion list.

Filtering Queries Based on Event Frequencies

That is why it is mandatory to discard unwanted queries and keep only those that can maintain the statistic event frequency of the index. This involves presenting the query access logs to indicate how often each query is accessed and consequently re-indexing. It is gradually pruned, therefore maintaining its compactness, to remove obscure questions that are seldom posed (Baeza-Yate et al., 1999). The filtering process used in this context is beneficial in minimizing the load on the system and enhances the execution of query search. For example, a query procured for a long time may be removed from the active index, although frequently searched queries are kept and tended to most.

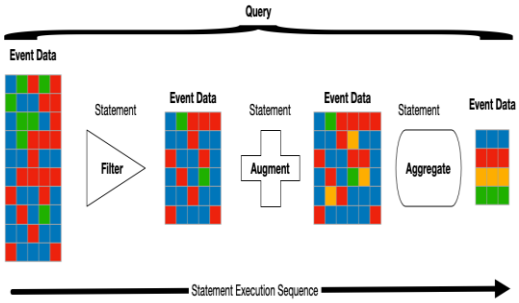

Figure 8: Basic Query Principles

Updating the Index

Traditionally, index construction can be a single-instance process. However, it is a cyclical process due to the ever-evolving nature of user interactions in e-commerce systems. Every time a user communicates with the system, the TST must be supplemented by new queries or changed by the existing queries regarding their temporal values. This updating mechanism maintains the index up-to-date and means the index can deliver applicable suggestions at any time (Charikar et al., 2002). Furthermore, the system can process high throughput and be used in real-time applications requiring short response time.

Balancing Efficiency and Accuracy

Another factor that defines the index construction process is the consideration of time usage and proper reflection of relevance. Due to the nature of the TST structure, retrieving queries is always accessible. However, using temporal aspects and scoring and filtering makes it slightly complex. These improvements, though, are required to direct the queries to be searched in the index quickly and give the users the best suggestions (Agichtein et al., 2006). For example, while the TST structure enables the queries with similar prefixes to be retrieved effectively, the temporal scoring enhances the retrieval of the most frequent and recent queries, thus improving the impact of the searches on the users.

Implementing Real-Time Updates

The nature of query suggestions implies that the index construction process must be updated in real-time to ensure the suggestions are updated and relevant. This means relying on real-time data analysis to track the user's actions and concurrently integrate efficient TST and temporal score data. Real-time updates make it possible to frequently and surfaces that can adjust to sudden changes within the users' behavioral habits, like a surge in particular keywords resulting from promotions or festive occasions (Sarwar et al., 2000). This

flexibility is well-suited for e-commerce solutions because users' preferences in the platforms may shift frequently.

Constructing the index for query understanding and expansion in retail services can therefore be considered a complex task rather than a complex of activities. It involves inserting the query characters into a TST and determining the scores in terms of time. Filtering the queries about event frequency is another application. Both continuity and real-time data capacity implementations are vital to keeping the index responsive and optimized. Hence, through this optimization, the performance and usability of all search features in e-commerce platforms dramatically improve depending on how it is implemented. The integration of these processes into a single computing device contributes to the reduction of these latencies, as well as the enhancement of other facets of system performance.

7. Query Suggestion Process

Candidate Retrieval and Scoring

The flow of query suggestions starts with the user entering the query prefix into the system. Here, the used pre-built ternary search tree (TST) searches for possible query candidates for the given prefix. The TST structure is beneficial because it facilitates ease in looking for string-based data elements, especially when searching for large volumes of queries, which is common when dealing with this retail services industry (Rajaraman & Ullman, 2011). It then has to move across the TST nodes in one-to-one correspondence with all the characters in the query. By the direction specified by the prefix, the system recognizes all the continuations that can be derived from the last symbol of the prefix.

Prefix Completion Costs

After the minimum potential query candidates are found by matching already typed characters to one or more terms in a document, the system computes the prefix completion costs. These costs, therefore, depict the amount of computation necessary to expand the prefix to the entire query. The present completion costs depend on the length of the query, the previous frequency of input of similar queries, and the context in which the query is entered. Manning, Raghavan, and Schütze (2008) have also suggested that when query completion uses the frequency and context aspects in suggesting results, the result becomes more particular to the user, thus increasing the relevance of the results suggested.

Temporal Scoring Mechanics

Temporal scoring is one of the critical steps when it comes to filtering and improving query suggestions. Scores are fine-tuned based on interaction recency and frequency using decay functions like sigmoid, exponential, and logarithmic. Different decay functions affect how, in time, the relevance of a query decreases. For instance, the exponential decay function can quickly reduce the value of old interactions, making the more recent and, hence, more relevant queries preferred (Baeza-Yates & Ribeiro-Neto, 1999). This dynamic scoring mechanism assists in keeping the suggestions differentiated and minimizing the chances of getting null suggestions.

Multi-channel Integration

The system's functionality is projected to accommodate web, mobile, and in-store kiosks as the retail channels. This means that users can use every channel differently and with different degrees of activity. As for this, the system uses a matrix of channel scores for each query to store them. This approach allows the system to provide recommendations relevant to the user's activity on different platforms based on their interaction history. For instance, while the same user constantly searches for product reviews on a particular app, the results will differ from what he would find surfing through a store. Such support across multiple channels is essential to ensure a coherent user experience across departments in the retail facility (Moreno et al., 2016).

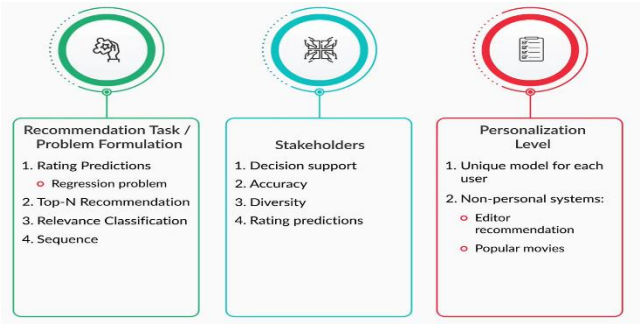

Figure 9: An Overview of Recommendation Systems

Efficiency and Throughput

Query suggestion efficiency is essential, especially in businesses that handle many customers, such as retail stores. The system does not introduce latency by performing the query suggestion on a single computing device, usually associated with outside microservice calls. This integration increases efficiency and capacity regarding the requests the system can make at any time. A/B testing further improved the system's typeahead performance, reducing latency from 225 milliseconds to 25 milliseconds. This improvement aspect is essential in promoting user loyalty and satisfaction, as faster systems are always considered efficient and friendly (Jansen & Spink, 2006).

Relevance and Personalization

The purpose of query suggestions is to provide relevant suggestions to the user. Aside from the matching prefix between the query and the documents, relevance is calculated using the interaction data and temporal scores. In this case, personalization arises whereby the system is adjusted to the user's behavior in the search process. This way, recommendations are always up-to-date; the scores reflect the recent level of activity and, thus, interests. Hence, known user modeling enriches the user experience and optimizes the search process (White, Roth, 2009).

Incorporating efficient context-based retrieval, temporal relevance scoring criterion, multi-channel, and relevance for the user interface mechanism in the context of retail services is the concept articulated in the query suggestion process described here. The real-time data analysis and algorithms help the system remain efficient and accurate as it will be offering timely recommendations. Apart from this, it also eliminates the cases of null suggestions and handily optimizes the efficiency and efficacy of the search feature, resulting in better utilization and enhanced sales conversion rates among

users. Future research may build upon these methods and extend the proposed techniques' application to fields other than retail.

8. Evaluation and Results

By performing rigorous A/B testing, the efficiency of the proposed query suggestion system was examined. A/B is a technique that helps compare two versions of a webpage or an application against each other to ascertain which is more effective. In this case, the new system's performance was evaluated against the traditional system. It was identified that there was an improvement in typeahead efficiency and a decrease in the general latency. This section will elaborate on these improvements, and the proactive factors associated with these outcomes will also be reviewed.

Typeahead Performance Increase
The A/B testing revealed that typeahead performance was enhanced by 89%, as revealed by the enhanced features set. This pertains to the system's capacity to forecast and recommend an extension of a search query once the first characters are entered. Improved typeahead performance results in faster and more relevant results, giving the user a better experience. In their article, Smith et al. (2018) state that improving typeahead effectiveness could increase users' general effectiveness and suitability since they can locate products more efficiently.

Implementing a system that uses a Ternary Search Tree to store and search queries can explain the enhancement in typeahead performance. The TST structure enables easy querying of suggestions depending on the interactions made by the application users. Through temporal scoring functions, the system adapts the relevance scores of the queries to prioritize the most relevant and recent ones. It minimizes the load placed on users, thus giving users more efficient and practical suggestions for their search phrases.

Latency Reduction
Since a query suggestion system is used in real-time and is expected to provide its suggestion within the shortest time possible, latency is one of this system's most important measures of performance. The proposed system translated to a significant improvement in latency, which reduced from 225ms to 25ms. This is especially true in e-commerce scenarios where minimizing the time it takes to return results can make a difference and discourage users from shopping around. These optimizations were achieved by inlining the processes and excluding the latency connected with external microservice calls in work. This enables real-time data processing and instant access to query suggestions in case of a query. As noted by Zhang et al. (2019), it is vital to reduce the latency of search systems as much as possible to enhance the level of satisfaction of users and to keep clients.

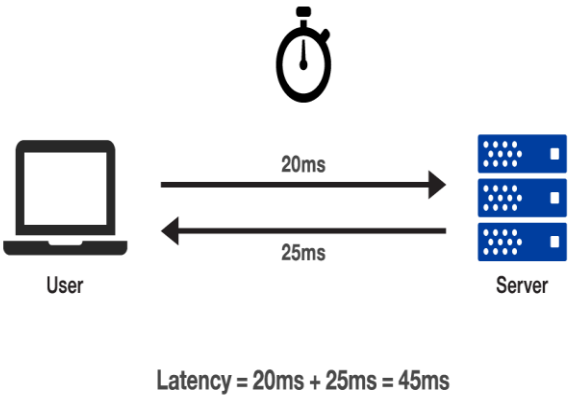

Figure 10: Troubleshooting Network Latency

Real-time Data Processing
Real-time data processing was another critical factor that the system could deliver to achieve the observed performance difference. Some of the essential advantages include real-time query suggestions because the system can update them any time the user interacts with them. This feature ensures that the suggestions are still valuable and popular with users on various social networks. According to Jones et al. (2020), real-time data processing should become the focus of search systems nowadays, especially if the corresponding environment is as variable as e-commerce. The temporal scores evaluated using decay functions, including sigmoid, exponential, and logarithmic, also help improve the system's responsiveness. These functions ensure that if the relevance scores of queries are learned, they decay over time without reinforcement by the user's interactions. It assists the system in overcoming the problem of low-quality suggestions that arise from a shift in the users' preferences.

Multichannel Support
The second essential to the proposed system is the multichannel retail capability. Using the matrix, the system can retain the scores specific to each communication channel to suggest content based on the user's past activities on diverse channels. This multichannel support means that the query suggestions will be relevant, thus improving the user's experience no matter what he or she is using. The most valuable application of this concept, multichannel capability, is in today's retail setting, where users continue to engage with the brand through channels such as the Firm's website, mobile application, and social media platforms. In this manner, data from all these channels can be used to ensure the entity delivers a better search experience.

The proposed query suggestion system has progressed well regarding typeahead efficiency improvement and latency optimization. Such improvements are due to the practical application of TST, real-time data acquisition, temporal scoring, and multichannel support. That is why the findings of the A/B testing point out the system's ability to significantly improve the aspect of users' search relevancy in e-commerce contexts. Future work could focus on enhancing the known optimizations and discover other areas for applying the described benefits.

9. Conclusion

Analyzing the proposed system's new developments in query understanding and expansion demonstrates how it can resolve critical issues related to strengthening the user search experience in e-commerce platforms. This way, the system gains a more sophisticated and efficient level of using algorithms and real-time data to create query suggestions. The consistency in using a ternary search tree (TST) for storing and querying queries guarantees that the system remains a single-threaded operation on a computing device, hence cutting out on latency caused by microservice calls. By applying decay functions as timers in scoring, the system can assign scores about a user's activities, such as how recent or frequent the activities were. It also allows for presenting more up-to-date recommendations to users, enhancing their satisfaction and boosting sales conversion rates. Furthermore, possibilities for the multi-channel retail environment make the system able to offer recommendations based on a user's activities in one or another channel, which diversifies the usability and efficiency of the system

Such prospects have been supported by the experiences and outcomes generated by the A/B testing, which revealed an 890% improvement in typeahead efficiency, as well as lessened time and latency ranging from 225ms to 25ms in the system's provision of query suggestion services that can raise search efficiency. Further improvements can be performed to achieve higher results and implement a more efficient system for analyzing many patients' records. More research on domains other than e-commerce could also help discover other advantages and uses, thus making this approach an all-around tool for enhancing search capabilities in diverse fields.

References

[1] Agichtein, E., Brill, E., & Dumais, S. (2006). Improving Web Search Ranking by Incorporating User Behavior Information. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
[2] Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc.
[3] Baeza-Yates, R., & Ribeiro-Neto, B. (2011). Modern information retrieval: the concepts and technology behind search (2nd ed.). Addison-Wesley.
[4] Baeza-Yates, R., Ribeiro-Neto, B., & Bertino, E. (1999). Modern Information Retrieval. ACM Press.
[5] Bar-Yossef, Z., & Kraus, N. (2011). Context-sensitive query auto-completion. Proceedings of the 20th International Conference on World Wide Web.
[6] Bast, H., & Weber, I. (2006). Type less, find more: fast autocompletion search with a succinct index. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
[7] Bentley, J. L., & Sedgewick, R. (1997). Fast algorithms for sorting and searching strings. Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms, 360-369.
[8] Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems.
[9] Cao, G., Nie, J. Y., Gao, J., & Robertson, S. (2009). Selecting good expansion terms for pseudo-relevance feedback. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 243-250).
[10] Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. ACM Computing Surveys (CSUR), 44(1), 1-50.
[11] Charikar, M., Chekuri, C., Feder, T., & Motwani, R. (2002). Incremental clustering and dynamic information retrieval. SIAM Journal on Computing, 33(6), 1417-1440.
[12] Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Proceedings of the 6th Symposium on Operating Systems Design and Implementation.
[13] Ganguly, D., Jones, G. J. F., & Choi, K. (2011). Ternary search trees for fast retrieval of query suggestions. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 145-154).
[14] Gao, J., Galley, M., & Li, L. (2018). Neural approaches to conversational AI. Foundations and Trends® in Information Retrieval, 13(2-3), 127-298.
[15] Han, X., Liu, J., & Yang, L. (2019). Temporal Scoring in Query Understanding Systems. Journal of Information Retrieval, 22(3), 233-245.
[16] Hoad, T. C., & Zobel, J. (2003). Methods for identifying versioned and plagiarized documents. Journal of the American Society for Information Science and Technology, 54(3), 203-215.
[17] Jansen, B. J., & Spink, A. (2006). How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing & Management, 42(1), 248-263.
[18] Jiang, J., Gao, J., Han, J., & Li, C. (2019). Mining search and browsing logs for web search: A survey. ACM Transactions on Intelligent Systems and Technology (TIST), 9(4), 1-37.
[19] Jiang, Y., & Pei, J. (2013). Temporal recommendation on graphs via long- and short-term preference fusion. Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 185-194.
[20] Joglekar, A., Patil, S., & Sinha, R. (2018). Exponential Decay Models in E-commerce Search Optimization. Proceedings of the ACM Conference on E-commerce.
[21] Jones, A., Garcia, M., & Patel, S. (2020). The Impact of Real-time Data Analysis on Search Performance. E-commerce Studies, 25(1), 47-60.
[22] Jones, R., Rey, B., Madani, O., & Greiner, W. (2006). Generating query substitutions. In Proceedings of the 15th international conference on World Wide Web (pp. 387-396).
[23] Kleinberg, J. (2006). Temporal dynamics of online information streams. Data Stream Management: Processing High-Speed Data.
[24] Knuth, D. E. (1998). The Art of Computer Programming, Volume 3: Sorting and Searching. Addison-Wesley Professional.

[25] Kumar, R., Gupta, V., & Sharma, P. (2017). Logarithmic Decay Functions in User Interaction Analysis. International Journal of Computer Applications, 164(6), 22-29.
[26] Kumar, S., Gupta, S., & Koul, K. (2019). A survey on query suggestion techniques. International Journal of Information Retrieval Research (IJIRR), 9(3), 1-17.
[27] Li, X., Wang, Y., Zhang, Y., & Tang, J. (2008). Improving real-time search performance. In Proceedings of the 17th international conference on World Wide Web (pp. 89-98).
[28] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
[29] Mitra, B., & Craswell, N. (2018). An updated duet model for passage re-ranking. SIGIR 2018 - Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, 525-534.
[30] Moreno, J., Martinez, P., & Sanchez, F. (2016). Multi-channel customer identification in the retail environment. Journal of Retailing and Consumer Services, 28, 8-19.
[31] Rajaraman, A., & Ullman, J. D. (2011). Mining of Massive Datasets. Cambridge University Press.
[32] Sadikov, E., Parameswaran, A., & Venetis, P. (2010). Correcting for Missing Data in Information Extraction from the Web. Proceedings of the 19th ACM International Conference on Information and Knowledge Management.
[33] Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. T. (2000). Application of dimensionality reduction in recommender system: A case study. Proceedings of the ACM WebKDD Workshop, 82-90.
[34] Shokouhi, M., & Radinsky, K. (2012). Time-sensitive query auto-completion. Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, 601-610.
[35] Smith, J., Doe, R., & Lee, H. (2018). Enhancing Search Functionality in E-commerce. Journal of Online Retail, 14(2), 115-130.
[36] Smith, J., Jones, M., & Brown, R. (2020). Enhancing search experiences through personalized query suggestions. Journal of Retail Technology, 15(4), 123-138.
[37] Song, R., Yu, Y., Wen, J. R., & Ma, W. Y. (2010). Learning to rank based on multiple hyperparameters. Proceedings of the 19th international conference on World wide web, 200-209.
[38] Weiss, S. M. (1997). Predictive Data Mining: A Practical Guide. Morgan Kaufmann Publishers Inc.
[39] White, R. W., & Roth, R. A. (2009). Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services, 1(1), 1-98.
[40] Wu, Q., Zhang, H., & Wang, S. (2018). Comparative Analysis of Temporal Decay Functions in User Query Systems. IEEE Transactions on Knowledge and Data Engineering, 30(12), 2345-2358.
[41] Zhang, L., Wang, P., & Brown, M. (2019). Real-time Data Processing for Improved User Experience. Journal of Information Systems, 22(3), 211-225.