Dynamic Query Substitution in fast evolving fashion

Kritika Jain
Myntra Designs Pvt Ltd
Bangalore, Karnataka, India
kritika.jain@myntra.com

Nilaksh Bajpai
Myntra Designs Pvt Ltd
Bangalore, Karnataka, India
nilaksh.bajpai@myntra.com

Ankul Batra
Myntra Designs Pvt Ltd
Bangalore, Karnataka, India
ankul.batra@myntra.com

Pajjuri Naveen Kumar Reddy
Myntra Designs Pvt Ltd
Bangalore, Karnataka, India
pajjuri.reddy@myntra.com

ABSTRACT

In most e-commerce platforms, search system supports free text keyword based queries. These queries can be very open ended. In contrast with traditional web search engines, recall set for e-commerce platform is usually constrained by the underlying inventory of products that are available in stock. This demand and supply gap leads to null or sub-optimal query results, which in turn, leads to a bad user experience. In this paper, we propose a query substitution approach based on semantic comprehension of keywords in fashion domain which improves query results while preserving primary intent of the user. Our solution enriches query results by selectively replacing query terms with hypernyms and co-hyponyms using Entity Affinity Relationship Graph (EARG). We demonstrate significant improvement in recall and overall products sold through various experiments conducted on a leading fashion e-commerce portal.

CCS CONCEPTS

• **Applied computing** → *Online shopping*;

KEYWORDS

E-commerce, Fashion, Query Suggestion

ACM Reference format:
Kritika Jain, Nilaksh Bajpai, Ankul Batra, and Pajjuri Naveen Kumar Reddy. 1997. Dynamic Query Substitution in fast evolving fashion. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK'97),* 7 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

In an online shopping portal, user query typically is a salad of words. A search engine is responsible to process this query and retrieve relevant results for the user to explore. E-commerce search is catalogue based and rely on precise descriptions of products. In contrast to web searches, zero result searches are more probable in e-commerce search as user queries are quiet descriptive and expected results are very specific. Also, recall set is constrained by the availability of products in stock.

Zero results leads to buyer frustration and potential loss in revenue. To counter this problem, numerous techniques have been developed which generate query suggestions on the basis of user click-stream and session data. Query reformulation or query rewriting is performed either when the user query is an imperfect description of the information needed or information retrieval engine fails to understand the query completely.

Null queries are most unique and most engines do not possess enough intelligence to handle them [5]. Query transformation or reformulation is seen as rewriting a query to retrieve a set of intended results. It can be achieved by query expansion and query substitution. Query expansion uses techniques like pseudo-relevance feedback and query relaxation or deleting query terms [3]. Query substitution uses content based understanding [1, 4, 8, 10]. These techniques modify the query to make it nearer to the content intended. Hence they are more relevant to web searches where the documents are elaborate and accurate mapping of user query to appropriate content is needed.

We define query substitution as an aid for the user to get the closest results when there is no matching content available for the exact query. We perform query Expansion by dropping irrelevant terms and query term substitutions using Entity Affinity Relationship Graphs to generate appropriate query suggestion.

Our information retrieval system is a search engine for fashion catalog with more than a million products. A user in any e-commerce platform either casually browses the catalogue or intends to purchase a targeted product. Search is an efficient for the latter. The technique discussed in this paper is specifically designed for improving search in fashion domain. We present a novel approach of query suggestion by query substitution pertaining to human fashion interpretation. While working in this domain, we deal with numerous complexities like the following:

(1) Highly dynamic vocabulary: Fashion changes in a rapid pace and so does the vocabulary that defines it. Various factors ranging from trending movies, actors or celebrities to climatic conditions, affect this change. Hence there is no fixed vocabulary that we deal with.
(2) Human purchase behavior: Even though user query depicts targeted fashion searches, they end-up buying close variants.

In: J. Degenhardt, G. Di Fabbrizio, S. Kallumadi, M. Kumar, Y.-C. Lin, A. Trotman, H. Zhao (eds.): Proceedings of the SIGIR 2018 eCom workshop, 12 July, 2018, Ann Arbor, Michigan, USA, published at http://ceur-ws.org

So it is required to transform the query while understanding preferences and intents of the users.

(3) Varied preferences for product attributes: Preference of attributes may differ in each case. For example, in the query "Nike black slip-on casual shoes", due to non-availability of exact product, we need to substitute or drop one or multiple attributes. The user might be more interested in shoes of brand "Nike" and can compromise with other product attributes, or he might be more interested in a black slip-on shoe, be it of any other brand. Evaluating the importance of each attribute contextually is a difficult problem.

Rest of the paper is organized as follows. We present the related work for query reformulation in the next section. We briefly describe how our search system works in section 3. Section 4 covers the architecture of Query Substitution framework followed by a detailed look at Entity Affinity Relationship Graph (EARG) in Section 5. Experiments and Results are explained in section 6.

2 RELATED WORK

Handling zero result queries to improve the recall of search system in e-commerce is explored by many researchers. Most of the work utilizes click-stream and user session data. In [5] time-based relevance feedback is used to improve the fidelity of rewrites. The algorithm performs query relaxation by searching the original query against a database of expired items, using the meta-data of matched expired items to constraint the rewritten query, and get more precise results while matching against a database of active items.

In [2] query is transformed using re-query feedback. User query reformulation activities in the form of deletion, insertion and modification of terms in the query are used to create term-transition graphs which in turn is used for suggesting. In contrast to this approach, we discuss a graphical representation of terms in the query and affinity relationships among them, extracted from the pair of attributes explored together in a session.

Another piece of work [7] discuss context aware query suggestions. Similar queries are grouped into concepts and suggestions are provided based on the concepts. However, researches have shown that [6] the zero recall queries are close to being unique. Even the most popular zero recall queries do not repeat more than tens of thousands of times within a month. But the most popular non-zero recall query repeats more than millions of times. Hence we cannot rely only on query specific models. The implicit preferences of users, what attributes are explored interchangeably, insights of domain from users activities are a few factors that can contribute in building a query suggestion system closer to how a user himself might think.

Any search system has endless data in the form of user queries and clicked results in a session. This data is an excellent source of information and can be used appropriately to accurately and efficiently retrieve relevant results[9, 10].

In this paper, we focus on improving the recall for zero result queries in fashion e-commerce using user click-stream and session data. Our contributions are:

(1) Entity Affinity Relationship Graph (EARG) having weighted intra-attribute relations globally, and in context of the primary intent. The attributes may be in or outside the catalogue taxonomy.
(2) Application of EARG in query suggestion system which performs query relaxation and query transformation depending upon the query.

3 OVERVIEW OF SEARCH SYSTEM

Input to the search system is user queries in the form of free text keywords, seeking fashion products to buy or explore. Even-though, the queries are often targeted at what the user is looking for, there is a tendency to make a purchase even if close matches or variants are presented. In such a scenario, it is important to correctly interpret user intents and preferences and present a suitable set of results. Consider the example,

"Nike black shoes without laces"

It appears that user here intends to buy a casual shoe which should not have laces and is black in color, preferably of brand Nike. There is a lot of information a user provides in this query. It is important to parse the query, understand it, disambiguate it, and then appropriately retrieve the results.

Figure 1 demonstrates the overview of our search system. The user query is parsed, sanitized, analyzed and annotated as a part of Query Processing. These annotations in the output annotated query after query processing are from a finite set of tags that categorize all fashion terms into types of products and various attributes. Unknown words are not annotated. Results are retrieved firing the annotated query. If we do not have sufficient number of results to display, we transform the query in query substitution engine and again execute the query. The new set of results are then displayed to the user.

Substitution algorithm ensures sufficient number of relevant results for the user. This algorithm is explained in the next section.

4 QUERY SUBSTITUTION ALGORITHM

Before elaborating upon the substitution, it is important to understand the structure of a user query. Avery user query has a primary intent. We define primary intent, as an entity which cannot be modified. In most cases, it is the article type or category which the user is looking for. In addition to this, an attribute set is present in that describes the primary intent. Primary intent also serves as the context for attributes while performing substitutions. In case, no primary intent is identified, substitutions are performed in global context. This will be elaborated in the following sections.

In the previous example, "Nike black shoes without laces", 'shoes' is the primary intent as it is clear that the user intends to buy nothing else but shoes. Other attributes describing 'shoes' can undergo modifications depending on their respective importance. Hence, first step for substitution is prioritizing these attributes on the basis of their popularity. If there is a primary intent present and is identified, attributes are prioritized on contextual popularity else global popularity. Here, brand 'Nike', color 'black', and attribute 'slip-ons' (annotation for 'without laces') are assigned priorities as per the general user preferences and tendencies to compromise certain attributes. We describe computation of priorities computations in the

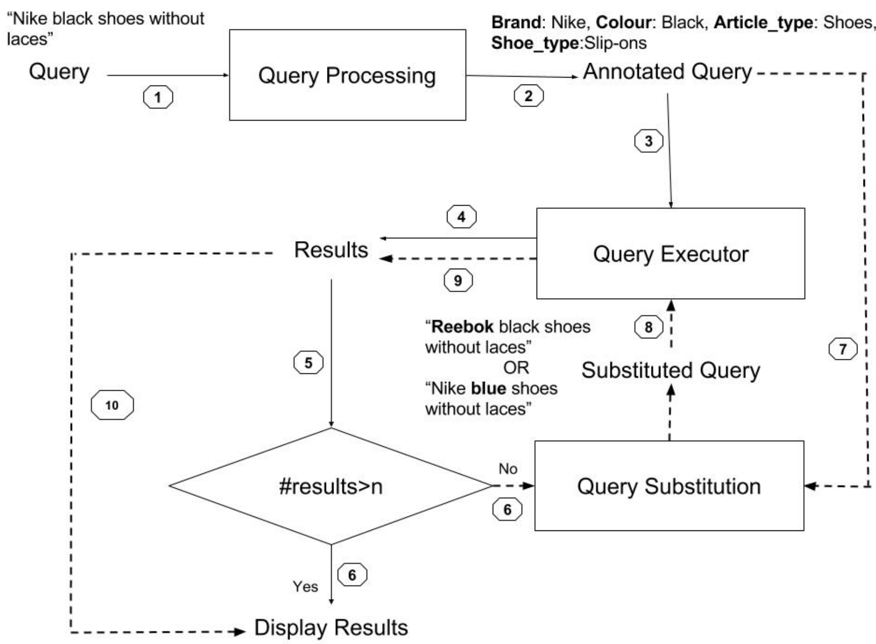

Figure 1: Overview of Search System

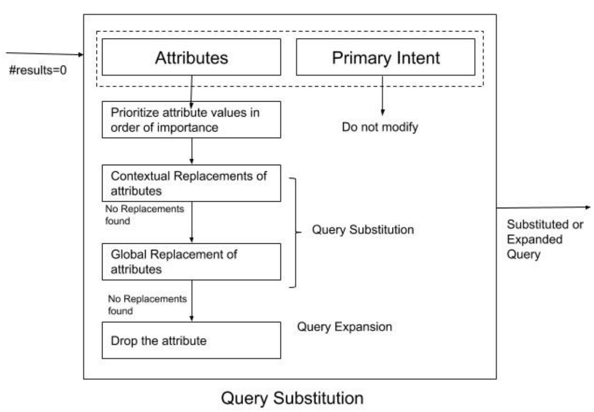

Figure 2: Query Substitution

next section. For the sake of understanding the algorithm, lets assume the Popularity(Nike)<Popularity(Black)<Popularity(slip-ons). As Figure 2 represents, how we substitute the terms in order of popularity and generate a list of substitutable queries.

Assuming that attribute Nike has least normalized popularity for shoes, we search for its substitutes in context of shoes. Substitution is performed in three simple steps:

(1) First round of substitution is performed in the context of primary intent when the attribute and primary intent have an association. For example, Nike sells shoes. I.e 'Nike' and 'Shoes' are associated. We extract the substitution of 'Nike' in context of 'shoes'.
(2) Second round of substitution is performed in the context of primary intent but there is no direct association of attribute to primary intent. For example, if 'Nike' does not offer shoes, we find a brand which is similar to 'Nike' but also offers shoes. This step ensures the preservation of implicit properties of attributes. If user searches "Reebok dress" the intent is to buy a sporty dress. If Reebok does not makes dresses, we find another sports brand which offers dresses and substitute the query.
(3) Third is global substitution. In absence of any query intent, global substitution is performed. Global substitution ignores 'Shoes' and looks for any brand closest to 'Nike'.

Outcome of this step is a list of substituted queries sorted in order of closeness to the original query. The top most candidate is picked from the list and result is displayed to the user.

Query Relaxation Vs Query Substitution

Query expansion is a fast and effective way of handling zero result queries. List of attributes are sorted in order of popularity and the least popular attribute is simply dropped. This is an ideal treatment for a low recall search system but at the same time compromises with relevance of products. Consider a query 'Nike skirts'. User here is looking for skirts which are sporty as 'Nike' is a brand, offering products mostly pertaining to sports. The explicit intent is a 'skirt' of Brand 'Nike', implicit intent is a 'skirt' used for sports.

Query Expansion will get rid of Brand 'Nike' and display all the skirts in the catalog. Query Substitution, in contrast, will look for

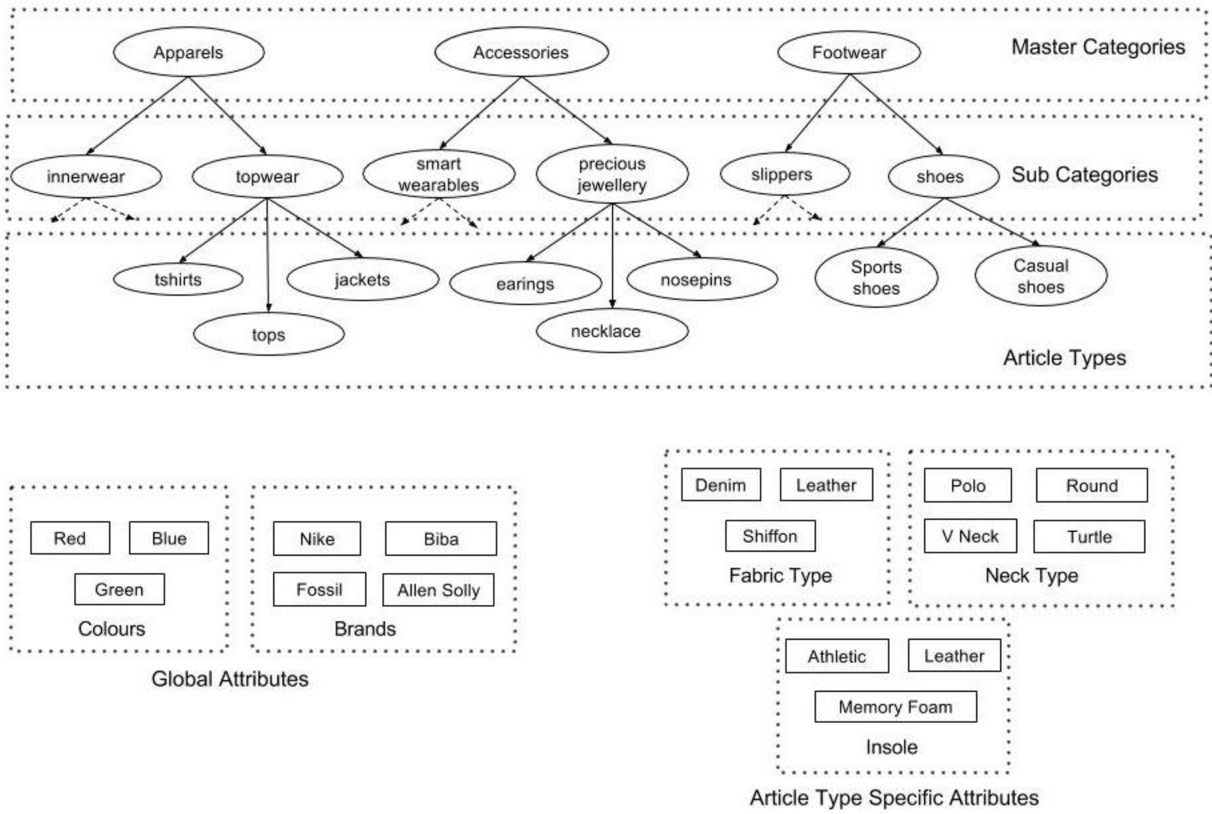

Figure 3: Graph Nodes

other brands of similar nature and show the results which are much more relevant.

A query transformation system which expands or substitutes appropriately is an ideal way to have a high recall with relevant results. The query should be expanded when the attributes are in no way relevant or do not help us to derive any hidden intent. Any attribute of value should be utilized to improve result relevance.

5 ENTITY AFFINITY RELATIONSHIP GRAPH (EARG)

Entity Affinity Relationship Graph is a weighted directional graph created using everyday user preferences while exploring products on the fashion portal. This section describes the domain, data used, and algorithm used to create these graphs.

5.1 Components of EARG

EARG is the representation of universal fashion entities, and the inter-relation between them, established from user sessions comprising of product exploration events.

Figure 3 shows the hierarchy of entity types in EARG. Nodes in the graph can broadly be categorized into 3 types. figure 3

(1) Product category or type: These form the primary intent of the query.
(2) Global Attributes: Product attributes applicable to all product categories and types. For example, brands, colors, usage etc.
(3) Article Type Specific Attributes: Product attributes specific to products. For example, Sole Type for Shoes, Neck and collar type for tops and t-shirts etc.

Each node is attached with a popularity score which is used to compare the importance of each attribute in a query when no primary intents are identified.

Edges represent inter-relationship among the nodes. There are 3 types of relationships that can exist:

(1) Article Type specific popularity:
These are weighted edges from global attributes and article type specific attributes to Product category nodes (master Category, sub-category and article types). Weights represent the popularity of global or article type specific attributes given the product category context to which the edge is connected.
(2) Global Affinity of attributes:
These are weighted edges between two nodes of same type. For example 'Nike' is connected to 'Reebok' as both are sports brands. Weights represent the strength of affinity.
(3) Article Type specific Affinity of attributes: These are weighted edges representing affinity between nodes, given any context (master Category, sub-category and article types)

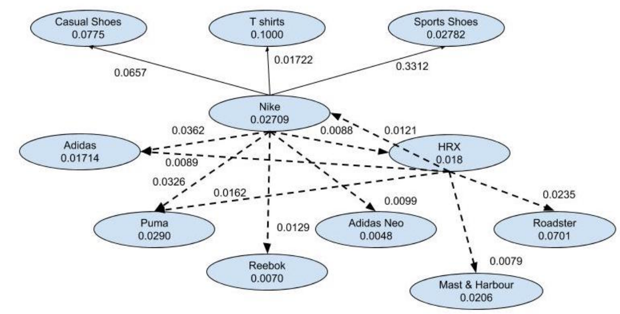

Figure 4: EARG sample representing Global Popularity, Article Type specific popularity and Global Affinity of attributes

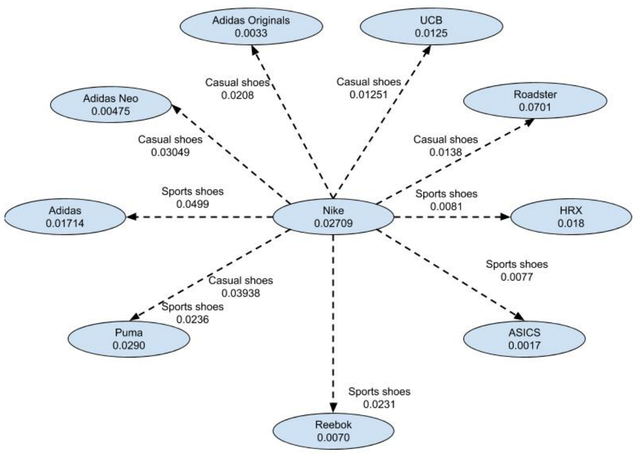

Figure 5: EARG sample representing Article Type specific Affinity of attributes

In Figure 4, solid arrow represents article specific Popularity. i.e. popularity of 'Nike' as a brand when article type is 'casual shoes' is 0.0657. The dashed arrows represent affinity between two attributes of same type. In this case, it is brand to brand affinity. The above diagram shows top 5 similar brands to 'Nike' and 'HRX', where weights on the edges is the affinity score. 'Nike' has maximum affinity to 'adidas' with a score of 0.0362 and least affinity to 'HRX' with a score of 0.0088. Similarly, 'HRX' has maximum affinity to 'Roadster' with score 0.0235 and least affinity to 'Mast & Harbour' with score 0.0079.

In Figure 5, dashed arrow represent top 5 Article specific affinity scores for Brand 'Nike'. For example, for article type 'casual shoes', Nike has maximum affinity to 'Puma' and minimum affinity to 'UCB' where as when the article type is 'sports shoes', 'adidas' is the most similar brand and 'ASICS' is least similar.

Every information captured in the above graph helps in making decisions during substitution of query terms. As described in the previous section, global popularity is used to prioritize attributes when no primary intent is found, Article type specific popularity is used otherwise. "Global affinity" and "Article type specific affinity" are used to find substitutions for attributes.

5.2 User Session Data

User session data is collected from the online e-commerce portal of Myntra Designs Private Limited. Every user exploring the fashion store is a part of a session. Within a session, multiple products are explored (clicks and product views). These products can be pivoted upon a single intended purchase. The user is taken from list page to product display page upon click on any specific product. This event is captured for every user. The basic and reasonable assumption here is that within a session, user clicks or views products which are similar to each other. This can help us in establishing the affinity relations and popularity of various products and its attributes. Session data needs to be pre-processed in order to get rid of numerous useless sessions.

(1) Sessions with a single product clicked are removed as it will have no information of interrelation within .
(2) A user might intend to purchase multiple article types in a single session. In such a case, the session is broken into multiple chunks of interrelated products grouped on article type.
(3) Product attributes of all the products clicked are collected.

5.3 Popularity and Affinity computation

Popularity of an attribute is the measure of its importance among other attributes of similar type.

Table 1: Brand vs Session counts for Article Type shoes

Brands/Session	S_1	S_2	S_3	S_4	S_5	Total products per Brand
Nike	10	2	0	8	5	25
Puma	20	3	8	2	4	37
Adidas	10	5	2	2	1	20
Forever21	2	0	1	6	0	9
Zara	6	0	1	0	8	15
Allen Soley	0	0	0	0	0	0
Total Products per session	48	10	12	18	18	106

$$Popularity(A_n) = \frac{\sum_{j \in session} \left(\frac{A_{nj}}{\sum_{i \in Attribute} A_{ij}} \right)}{\#session}$$

Here, $Popularity(A_n)$ is the popularity value of a given attribute A_n. 'session', as described above is a set of consecutive searches by a particular user for a given article type. A_{ij} is the number of times attribute A_i is viewed in j^{th} session. $Attribute$ is a set of all possible values of attributes of same type.

Table 1 is a sample data for a few brands explored for 'shoes' by various users in the respective sessions. Article type specific popularity of Nike, $Popularity(Brand_{Nike})$, for when user is looking for shoes, can be computed as follows:

$$Popularity(Brand_{Nike}) = \frac{\frac{10}{48}+\frac{2}{10}+\frac{0}{12}+\frac{8}{18}+\frac{5}{18}}{5}$$

To compute the global popularity of $Brand_{Nike}$, similar computations will be done across all the products, irrespective of what is the article type.

Affinity between two attributes represents how confidently we can substitute one attribute with another attribute of the same type in a global context or for a given article type. Note that this is not similarity between attributes, but what is the possibility that a user looking for one attribute can compromise and go for another attribute.

$$Affinity_{A_n \to A_m} = \mu_{1/2}\left(\frac{A_{n1}}{A_{n1}+A_{m1}}, \frac{A_{n2}}{A_{n2}+A_{m2}}, ..., \frac{A_{ni}}{A_{ni}+A_{mi}}\right)$$

Here, $Affinity_{A_n \to A_m}$ represents the Affinity of an attribute A_n to attribute A_m. $sessionWithA_m > 0$ are the sessions where A_m is explored at least once. $\mu_{1/2}$ is the median of values from all the sessions. We prefer median over mean as the data can be skewed depending on the availability of products.

Table 2: Affinity of multiple brands to Nike

Brands (B_i)	$Affinity_{B_i \to Nike}$
Puma	0.6
Adidas	0.5
Forever21	0.16
Zara	0.375
Allen Solly	0

Similar to Global Popularity, the affinity between attributes in a global context is computed across all article types.

6 EXPERIMENTS AND RESULTS

We measured the performance of query substitution basis product discovery and products sold through typed search for a static cohort of 1Mn users on a e-commerce platform of Myntra Designs Pvt Ltd. Experiment base was divided into an equal number of users in test and control groups. Our new method(test) was compared with the fashion agnostic query expansion(control).

The experiment was repeated for three weeks to confirm the performance of the method. The statistical significance of the results was measured by p-value of a test on null hypothesis H_0.

H_0: *The difference in metrics between test and control groups is caused by random variation.*

Product discovery is measured basis clicks on catalog exposed through search while keeping a close eye on recall basis zero results. We observed following improvements between the two groups:

(1) Percentage of catalog clicked through search
(2) Zero results queries

The improvement in product discovery basis increase in the percentage of catalog clicks from search is significant over status-quo methods of fashion agnostic query expansion. The recall basis zero-results query did not decline significantly. Moreover, we gained a 99% significance in catalog click improvements and 95% significance in increase in zero results. This resulted in 292K unique products clicked in test Vs 273K unique products clicked in control in week 1, 307K in test Vs 289K in control in week 2 and 293K in test Vs 275K in control in week 3.

Table 3: Percentage of catalog clicked through search

Week index	Test vs control improvements	Significance (p-value)
Week 1	7.15%	0.0021
Week 2	6.23%	0.0052
Week 3	6.49%	0.0089

Table 4: Zero results queries

Week index	Test vs control difference	Significance (p-value)
Week 1	-3.1%	0.038
Week 2	-2.2%	0.042
Week 3	-1.9%	0.029

To measure the impact of the trade-off between recall and product clicks, the percentage of products sold through search was used. It kept a check on the relevance of the results. Final purchase by customers is the biggest testament to the fact that they found what they were looking for. This is a key criterion for query substitution to succeed and suggest that the search results presented are actual substitutes for user queries.

Table 5: Percentage of products sold from Search

Week index	Test vs control improvement	Significance (p-value)
Week 1	0.39	0.0064
Week 2	0.34	0.0032
Week 3	0.31	0.0041

Percentage of products sold through our method were at-least 0.31 percentage points higher than the existing query expansion approach. Query expansion approach is a safer approach but it can reduce the relevance and subsequently the products sold. Clearly, fashion aware query substitution ensures the high quality of search results while maintaining a high sense of recall in the system.

Analysis

To assess the quality of substitution, zero result queries in catalog search were collected for a duration of eight hours, with no query

Table 6: Examples of Query Transformation

Original Query	**Transformed Query**
michael kors leather bags	*fossil leather bags*
aldo leather bags	*aldo synthetic leather bags*
nike black rf cap	*nike black caps*
kenneth cole watches	*tommy hilfiger watches*
elle handbags	*forever 21 handbags*

substitution. Theses queries were then re-executed in an identical offline setup with query substitution on. 61% of these queries successfully retrieve relevant results. Few examples of Query transformation seen in this experiment, in the form of expansion as well as substitution are shown in table 6.

Table 6 clearly shows how a brand is substituted with a similar brand within the respective contexts. In the first example, 'Fossil' is the brand where the user finds leather bags similar to what they get in 'Michael Kors' where as for someone looking for "Elle handbags", 'Forever 21' is the brand where the user is able to find what is needed. In another example, user is looking for 'aldo leather bags' which are not available and we show 'aldo synthetic leather bags' as they are not currently available in inventory.

Analysis of the misses in the experiment shows that a significant portion of these queries are either inappropriate for fashion domain (for e.g. "fidget spinner","red chaos", "vegetable cutter" etc.) or are extremely miss-constructed for the system to understand. In such cases, zero result is the most relevant result.

7 APPLICATIONS

Next best results have an extensive use of query term substitutions. The user query is substituted when we do not get any results from catalog to show to the user. In such a case the extent of query modification should be limited. Contrary to this, if exact product match to user's query has insufficient results or the modified and original query are different beyond a threshold, multiple query terms are substituted to create a list of next best results for the users to optionally pick any query as per their preferences.

8 CONCLUSION

We have presented an ingenious approach, dynamic query substitution, to improve the search results in the context of fashion catalog. Our method identifies a primary intent of the user and pivots on this core idea to provide substitute products. The method also ensures to capture all the information assessed from user query instead of unintentional dilution of information in other methods like query expansion. The candidate terms for substitution are identified basis EARG relations that capture the users' intent over multiple sessions. We have shown through experiments on a large user base that this method is significantly effective than the existing methods in improving the clicks and sell through of products. Dynamic query substitution has many practical applications as well such as showing the next best results in absence of enough exact matching products in fashion e-commerce.

REFERENCES

[1] Yigal Arens, Craig A Knoblock, and Wei-Min Shen. 1996. Query reformulation for dynamic information integration. *Journal of Intelligent Information Systems* 6, 2-3 (1996), 99–130.

[2] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 875–883.

[3] Rosie Jones and Daniel C Fain. 2003. Query word deletion prediction. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 435–436.

[4] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*. ACM, 387–396.

[5] 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*. ACM, 73–82.

[6] 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*. ACM, 75–84.

[7] Yang Song, Dengyong Zhou, and Li-wei He. 2012. Query suggestion by constructing term-transition graphs. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 353–362.

[8] Egidio Terra and Charles LA Clarke. 2004. Scoring missing terms in information retrieval tasks. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 50–58.

[9] Xuanhui Wang and ChengXiang Zhai. 2007. Learn from web search logs to organize search results. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 87–94.

[10] Xuanhui Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 479–488.