Netcore UNBXD

Intent Search

Netcore Unbxd White Paper 2024

unbxd.com

Introduction to

Netcore Unbxd Intent Search

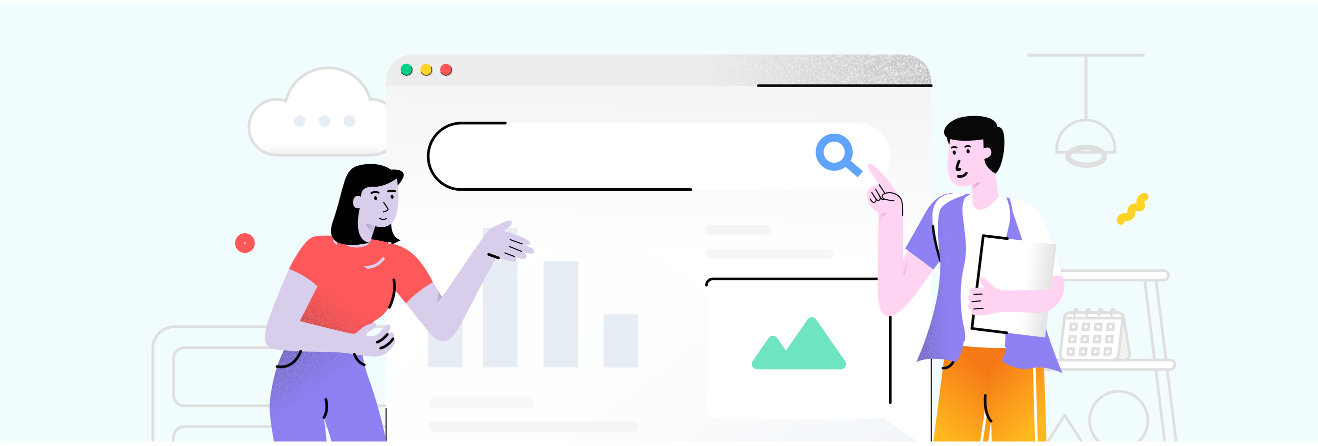
In the world of eCommerce, search functionality plays a crucial role in helping users find the products they're looking for quickly and easily. However, traditional keyword-based search methods can fall short when understanding the nuances of user intent. This shortcoming can make extracting products from massive catalog listings feel like finding a needle in a haystack.

Netcore Unbxd Intent Search addresses this problem by combining Vector models and NLP or keyword-based Search within a **Neural Network**. It utilizes advanced machine learning and deep learning models to provide an efficient and effective solution for searching and retrieving contextually relevant information from large datasets. Intent Search emerges as a future-proof technology that overcomes the limitations of traditional approaches, ensuring a more precise, comprehensive, and scalable search experience.

In this whitepaper, we'll explore the depths of Intent Search, delving into its working, applications, impact, and how Netcore Unbxd implements it to redefine search capabilities.

Let's begin with understanding Vector Search

Vector Search serves as the foundation of Unbxd's Intent Search and brings a paradigm shift to information retrieval.



Vector Search is a semantic data retrieval technique that uses multi-dimensional spaces to plot and index unstructured data, such as text, images, audio, and video, to match with search queries.

Vector Search assumes that items with similar characteristics will have similar vectors in the multi-dimensional space, also called the Vector Space. That is, the closer the vectors are to each other, the more similar the items will likely be (using algorithms like cosine similarity and Euclidean distance). By representing data as vectors in a multi-dimensional space, Vector Search can understand semantic relationships and similarities between items. This semantic understanding enhances search accuracy and improves the overall user experience.

Vector Search: The Process

Vector Representation of Data

Vectors are objects used to represent data points in a multi-dimensional space (neural framework), where each dimension corresponds to a feature or attribute of the data point.

To understand this better, let's take a small data set of **four** shoes in a catalog with six attributes: **Brand, Price, Color, Size, Style,** and **Weight**.

The dataset will look like this.

Shoe 1

Brand: Nike
Price: $80
Color: Black
Size: 9
Style: Running shoe
Weight: Light

Shoe 2

Brand: Vans
Price: $60
Color: Black and white checkerboard
Size: 8
Style: Skate shoe
Weight: Light

Shoe 3

Brand: Converse
Price: $55
Color: Red
Size: 9.5
Style: Casual sneaker
Weight: Moderate

Shoe 4

Brand: Dr. Martens
Price: $120
Color: Dark Brown
Size: 10
Style: Combat boot
Weight: Heavy

Here, Shoe 1, Shoe 2, Shoe 3, and Shoe 4 are Vectors. The attributes of the product will be the Dimensions (like the x-axis and y-axis, but 6 of them) of the vector. In this example, each vector has six dimensions—Brand, Price, Color, Size, Style, and Weight.

We plot each shoe in vector space by assigning a numerical value to each dimension—Vectorization. Continuing with our Shoe dataset,

- **Brand**: Nike = 1, Vans = 2, Converse = 3, Dr. Martens = 4
- **Price**: $80 = 80, $60 = 60, $55 = 55, $120 = 120
- **Color**: Black = 1, Black and white checkerboard = 2, Red = 3, Dark Brown = 4
- **Size**: 9 = 9, 8 = 8, 9.5 = 9.5, 10 = 10
- **Style**: Running shoe = 1, Skate shoe = 2, Casual sneaker = 3, Combat boot = 4
- **Weight**: Light = 1, Moderate = 2, Heavy =3

Using this, we can map each shoe as a vector in a six-dimensional vector space, each dimension representing one of the shoe's attributes. In our example, the Vectorized values of our shoe dataset would be:

Shoe 1

{1, 80, 1, 9, 1, 1}

Shoe 2

{2, 60, 2, 8, 2, 1}

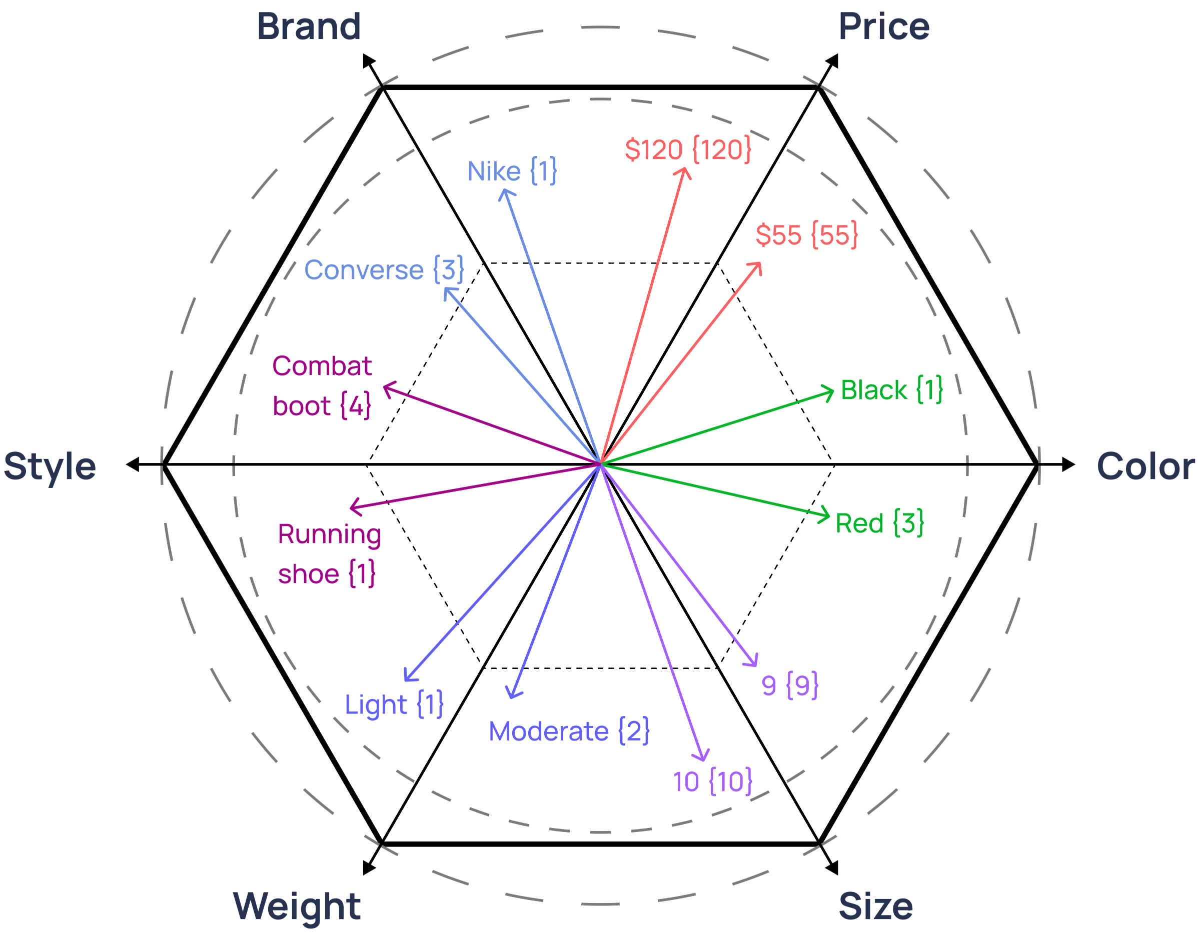
Shoe 3

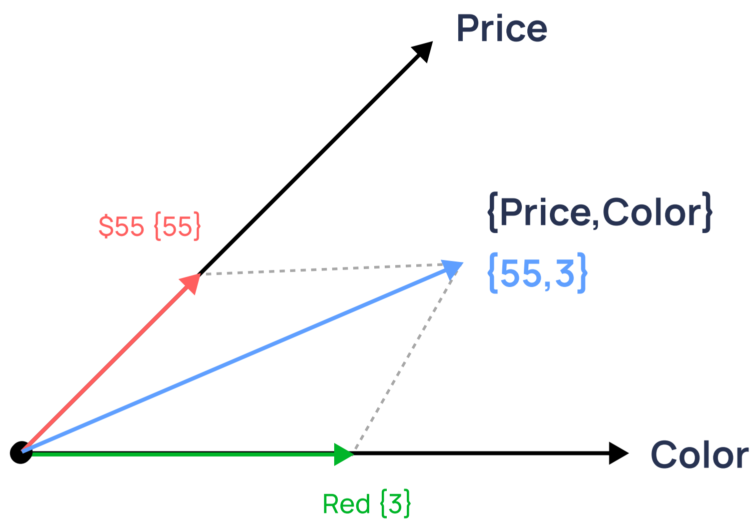
{3, 55, 3, 9.5, 3, 2}

Shoe 4

{4, 120, 4, 10, 4, 3}

The final dataset in the vector space would look like this:



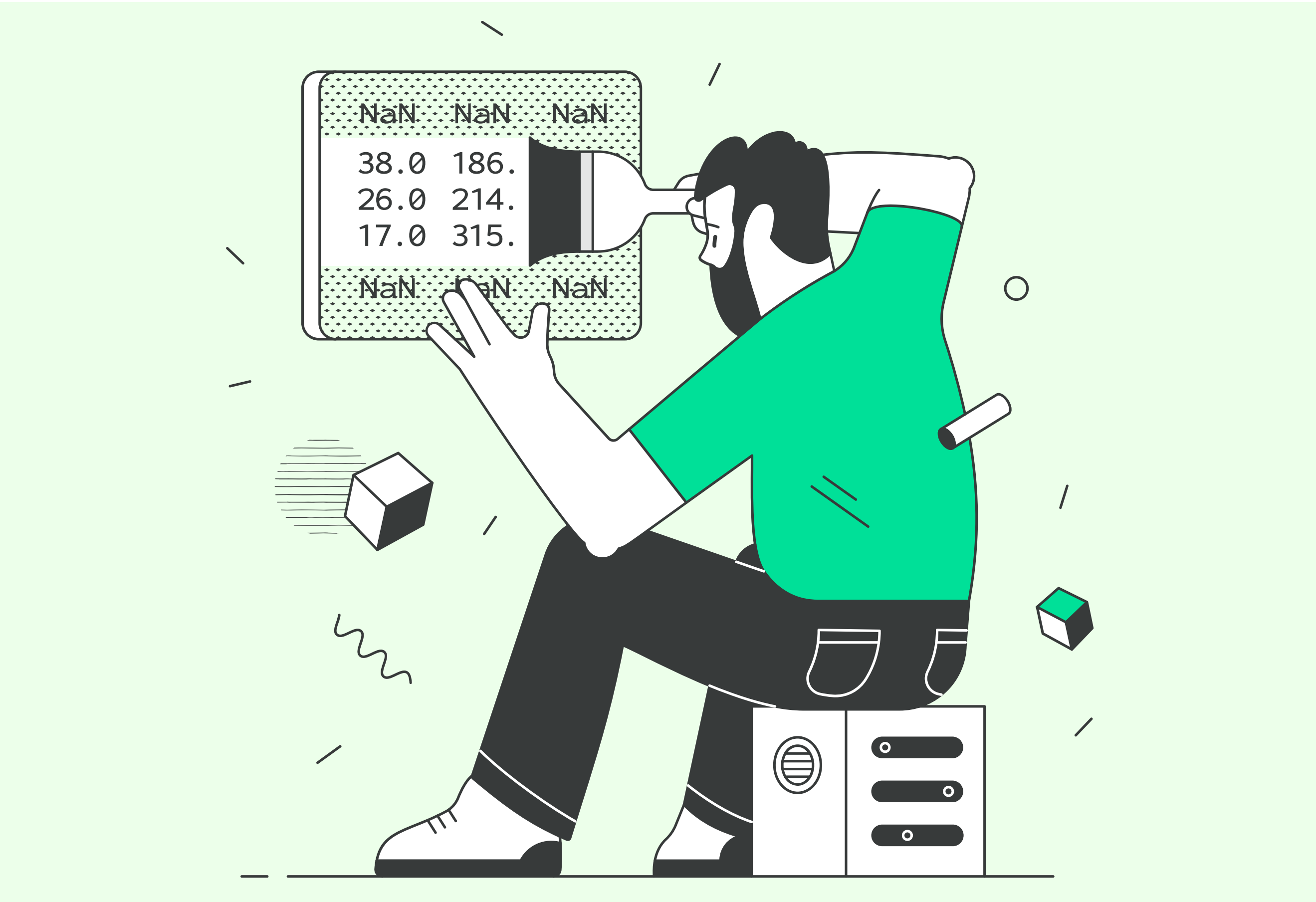


*Note: At Netcore Unbxd, we can go as high as representing **1024 or 1536 dimensions** in the vector space.

Indexing in the
Neural framework

The vectorized values of the dataset get indexed and are stored in a database for efficient data retrieval. Many traditional databases, like Vespa by Yahoo, were built for this purpose.

At Unbxd, we go one step further and fine-tune our Intent Search by creating shopper-specific and vertical-specific models using behavioral clickstream data. These models understand the nuances of industry-specific search queries and converts them into a vector representation.



Query Processing

You might wonder how plotting these Vectors and creating vector space help get highly relevant results for search queries. Well, here's where it gets interesting.

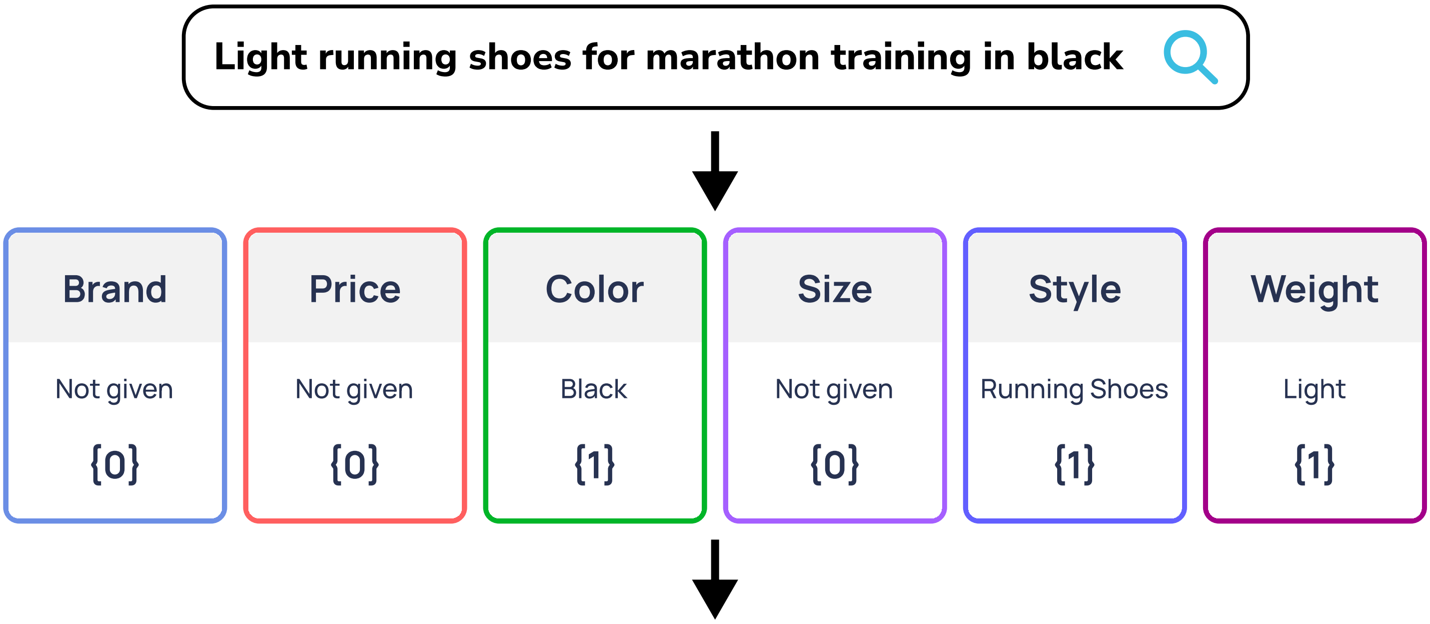
When a shopper puts forward a search query, it is converted into a vector, and the relevant dimensions are extracted. These dimensions go through vectorization using the indexed values of the dataset. The 'search query vector' is now plotted in the existing vector space, and the nearest neighbors with the highest similarities will be fetched as the search result.

For example, consider a user searching for "Light running shoes for marathon training in black." The traditional keyword-based search might match this query with products containing the words "running shoes." However, we can leverage Vector Search to understand the context and semantics of the query.

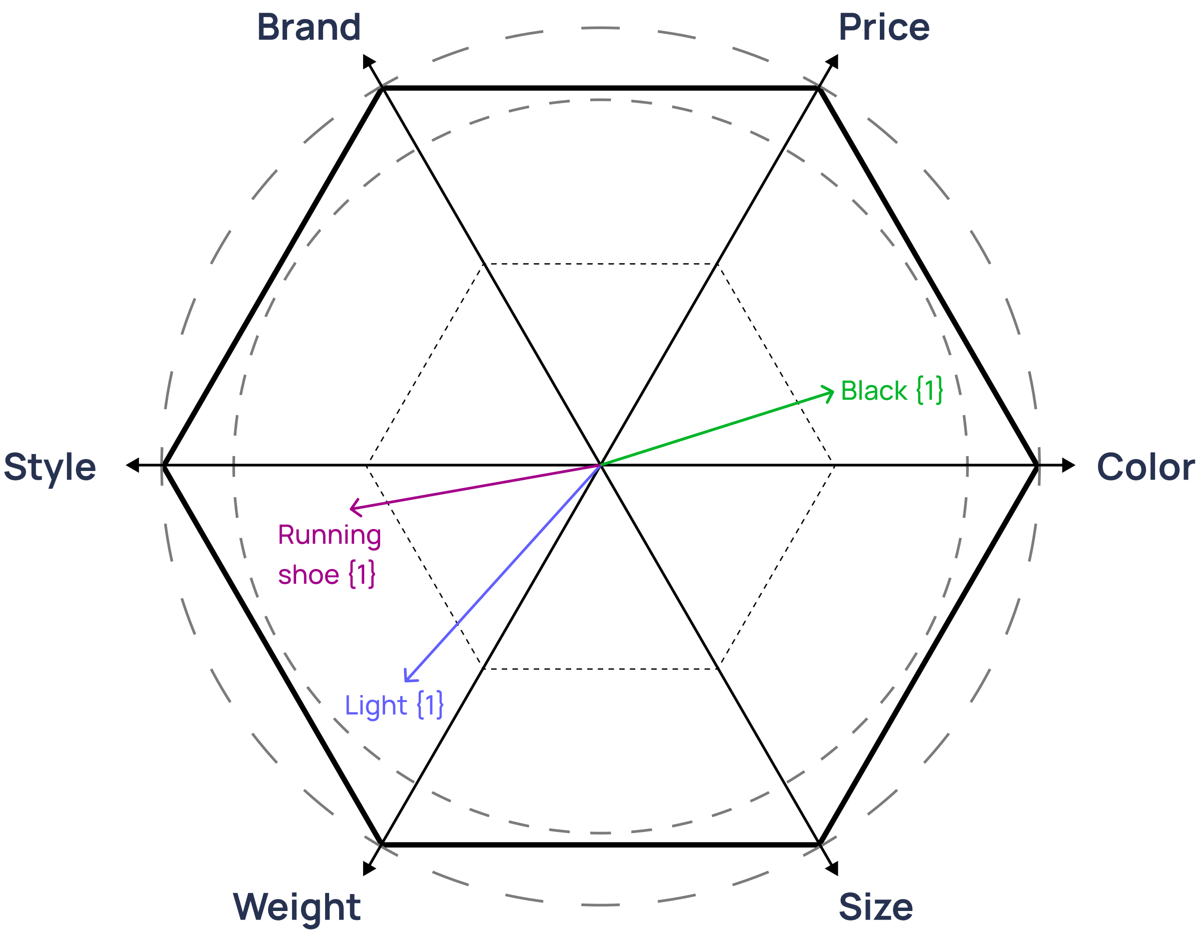
By analyzing the vector representations of both the query and the product catalog, we can identify running shoes specifically designed for marathon training, considering factors like comfort, durability, and performance.

In our example, the vector for the search query "**Light running shoes for marathon training in black**" would be (0, 0, 1, 0, 1, 1). The distance between the vector of the search query and each shoe in the multi-dimensional neural framework is calculated to find relevant results. First, the shoe vectors within a certain radius of the search query vector will be retrieved. Then, the retrieved vectors are ranked and filtered based on various criteria to provide results of high relevance.

For our example query, the top-ranking relevant result would be Shoe 1, a black Nike running shoe.



{0,0,1,0,1,1} = black Nike running shoe



How is Traditional (BM25) Search Different from Vector Search?

BM25 (Best Match 25) is a traditional text retrieval function often used for information retrieval and natural language processing tasks.

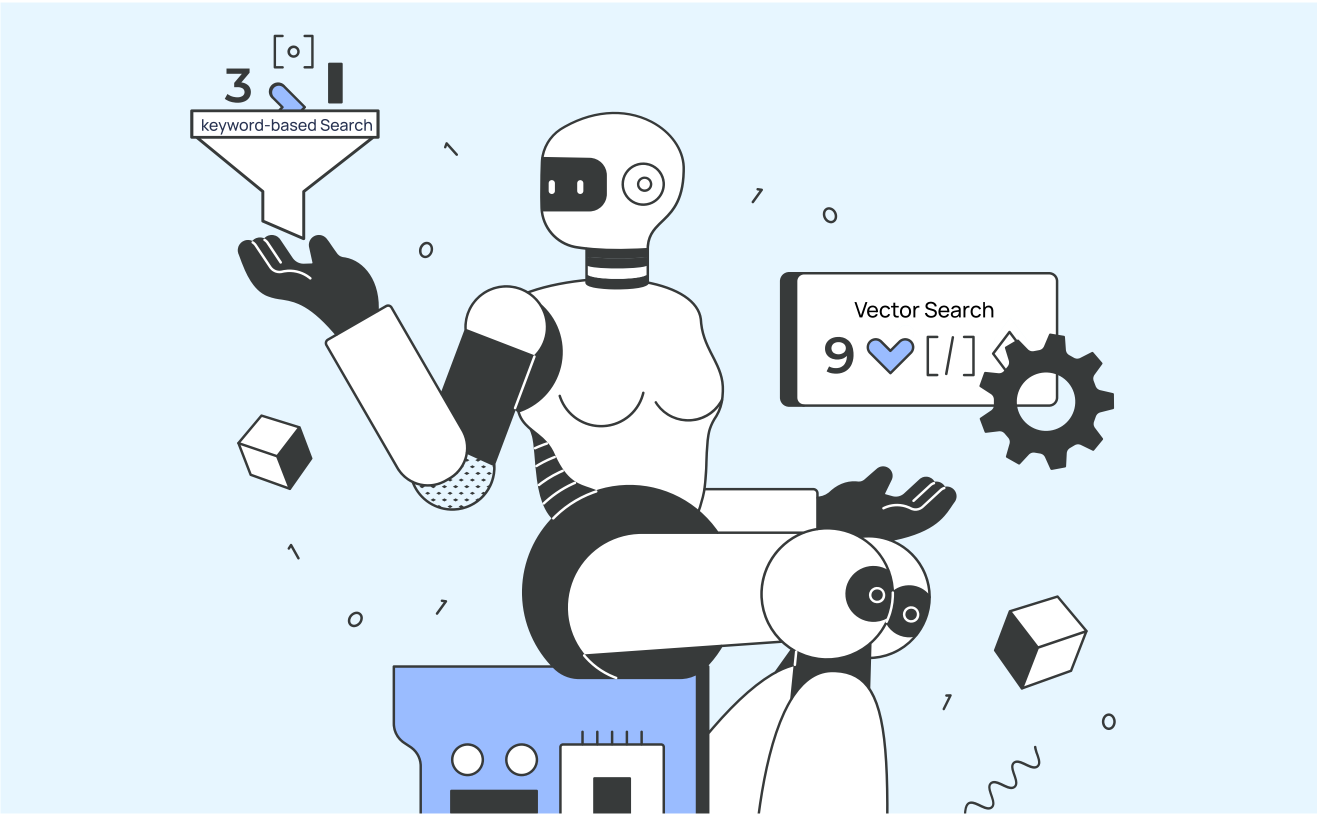
The traditional search uses boolean retrieval to match documents from the index. It is based on the assumption that relevance is proportional to the term frequency (the number of times a term appears in a document) and inverse document frequency (the rarity of a term across the entire corpus of documents). BM25 scores are then used to rank the documents in a corpus, with the highest-scoring items considered the most relevant.

On the other hand, vector search is based on representing products and queries as numerical vectors in a multi-dimensional space. Trey Grainger has described the core principle at work as "**a word is known by the company it keeps.**"

Vector-based semantic search can search not just on the term but also consider the context in which the term appears. The terms in the documents are encoded into n-dimensional vectors using a transformer-encoder and then indexed into a vector indexing database. At query time, the query terms also get encoded into vectors. A nearest neighbor search is performed between the query and document vectors to fetch the most relevant documents using a distance calculation metric such as cosine similarity. The documents with vectors closest to the query vector are returned. Vector search is typically more effective than BM25 when matching semantically similar phrases and dealing with synonyms and polysemy.

Intent Search

Vector searches, while extremely beneficial for long and heavily contextualized queries, are not always necessary when queries are short and to the point, such as "blue captain america T Shirt." It also comes with challenges like high computing costs.



Intent Search combines both Vector and NLP or keyword-based Search within a neural network such that it can adapt. By analyzing vectors, Intent Search identifies similarities, allowing for more accurate matching of queries to relevant products. For shorter queries, the system seamlessly switches to keyword-based search, ensuring speed and efficiency.

How does Netcore Unbxd Intent Search work?

Netcore Unbxd Intent Search represents an innovative approach that strikes a balance between precision and recall.

Textual search augmented by capabilities such as NER has been a differentiating forte demonstrated by Netcore Unbxd over the long years, the following describes the workings other half of intent Search that is Vector Search:

For a brand being powered by Netcore Unbxd, a neural network is built on vector representations for various searchable fields from the brand's product catalog. Like explained above when a shopper searches for a query, the query in itself is vectorised.

The Field vectors are in-turn used to calculate the similarity by matching query vectors against each of the field vector indices using ANN (approximate nearest neighbor) search and aggregate vector match score is calculated which is used as final match score to provide relevant results.

Advantages of ANN search:

Fast retrieval:
ANN search algorithms provide quick approximate nearest neighbor results, even in high-dimensional spaces like 1024/1536 dimension embedding we use.

Scalability:
ANN search methods are designed to handle large datasets efficiently.

Trade-off between accuracy and efficiency:
ANN search allows for balancing search accuracy with query latency.

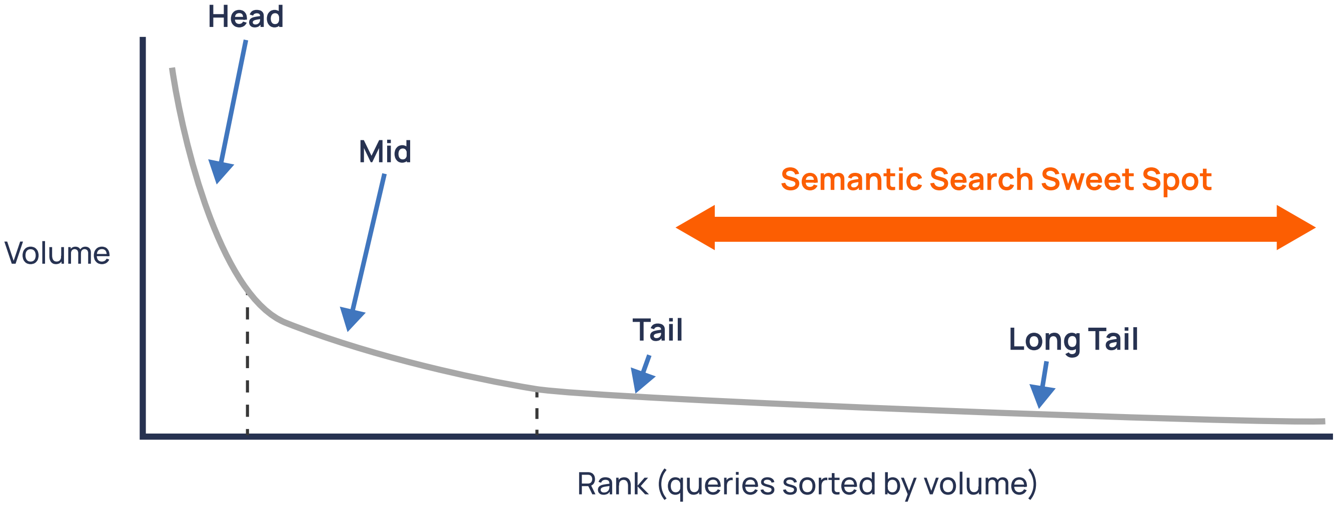
ANN Search strategy:
ANN search methods like k-d trees, locality-sensitive hashing (LSH), or graph-based approaches are chosen based on the dataset to optimize for retrieval/scalability. Section of the appropriate algorithm and tuning the parameters is done in the model tuning process.

Let us now take a look at how Netcore Unbxd helps us in the eCommerce industry:

Long tail query:
Query statistics across customer sites show that the most frequently searched terms that fall in the “head” and “mid” query category range are served well by traditional search strategies such as TF-IDF. Combined with user behavior data, they yield highly accurate results. However, when it comes to surfacing documents for less searched, “tail” and “long-tail” queries, these strategies result in decreased precision and recall.

This difficulty is augmented by the fact that there’s often no user behavior data for such queries. Sometimes, the recall can go down to zero even when there are products in the catalog that match the context of the query. This is because term matching can no longer derive context from the set of words in the query. The correct results in such a scenario are semantically similar to the query, but there’s no exact match.

Netcore Unbxd would then compare this vector to the vector representations of each document in the product feed and generate results based on similarity.



NLP search:
Natural language pattern search involves understanding multiple contexts in a single query and returning results closest to what the user is looking for. Here is a query from one of the prominent fashion brands in the US:

"High-waisted luxe comfort knit columnist pants."

Here, the user is talking about the pants they are looking for, which are "**High-waisted.**" Traditional search returns one result, which isn't the most relevant product for this query. According to our experiments, the most pertinent results are produced by vector search.

Query containing negative keywords:
Let's say a retailer is selling a "**cordless circular saw,**" and the user is typing "circular saw without cord." First, it is important to mine the hard negatives from the query and show the relevant results. This would require a lot of work to perform using traditional frequency-based search engines.

Question and answer search (conversational search):
These are typical queries that would be much more relevant when there is a blog search or search for an education platform. Google's Talk To Books is one example of how they have answered the question via the book's data.

Multi-language Search:
Intent Search allows text in different languages (100 plus) to be represented as vectors in a common high-dimensional space, so that vectors represent similar meanings regardless of language. When a user enters a query in one language, we can return results to the user based on the query's mapping to the vector generated above. This may provide a more accurate and comprehensive search experience.

A World beyond **Semantic Search**

A world beyond semantic search would involve using more advanced techniques and technologies to understand and process text data.

This could include:

- **Deep learning techniques** such as Recurrent Neural Networks (RNNs) and transformer models to better understand the context and meaning of text data.
- **Knowledge graphs** - a representation of data in the form of entities and relationships, which can be used better to understand the meaning and context of text data.
- **Semantic Role Labeling (SRL)** is a technique used to identify the roles and relationships between words in a sentence, which can be used better to understand the meaning and context of text data.
- **Multi-modal search** - a search that can process multiple types of data, such as text, images, and audio, to provide more accurate and relevant results.
- **Personalization and adaptation** - a search that can adapt to the user's needs and preferences, providing more accurate and relevant results.

These advanced techniques and technologies enable a more accurate and complete understanding of text data, which leads to more accurate and relevant search results and a better user experience.

Intent Search is a powerful and versatile technique for information retrieval and natural language processing. With its ability to handle large volumes of data and scale well, it's no surprise that many companies are investing in this area. However, it's important to note that this is not a one-size-fits-all solution and may not be the best option for every use case. Before investing, evaluating whether it aligns with your business needs and if the benefits outweigh the costs is essential. With the ongoing research and advancements in the field, the future is promising and holds great potential for organizations looking to improve their search capabilities.

Netcore
UNBXD

1710 S. Amphlett Blvd
Suite 124 San Mateo,
CA 94402

sales@unbxd.com
support@unbxd.com

+1 (650) 282-5788