pureinsights™

Discovery Platform

Tech Brief

A Platform for Building an Advanced Google-Like Search Experience

September 2022

Contents

Introduction

"How old is the moon?" is a question children used to ask their parents. But today, that child is just as likely to ask Google for the answer. And when that child is older, he or she might ask Google "What is the age of the actor who plays Neo's girlfriend?"

The answers from Google, respectively, are "4.53 billion years" and "55 years old".

Our use of Google and internet search is so prevalent that we don't even stop anymore to consider how Google continues to redefine what users expect from search. Nor do we reflect on what Google has done to enhance keyword search and turn Google into a question answering system.

Take for example the second query: "What is the age of the actor who plays Neo's girlfriend." In order to answer this question Google had to:

- Understand a query typed in natural language
- Determine that the question is about a character named "Neo" in a movie
- Surmise that the movie is "The Matrix"
- Determine that Neo's girlfriend is named "Trinity," played by actor Carrie-Ann Moss
- Calculate Carrie-Ann Moss's current age (depending on the year the query was made)

This is what is means when search application users say, "just make it work like Google."

This may seem daunting if your current enterprise or website search application barely gets by on mediocre keyword search. But the good news is that the technology behind a Google-like search experience is available today for your own search use cases. And much of it is open-source and compatible with today's modern cloud-based architectures.

The Pureinsights Discovery Platform™ (PDP) is a search application platform that integrates search technologies such as content ingestion, processing and indexing with technologies like knowledge graphs, machine learning and advanced natural language processing to deliver the search experience users expect today in an extremely cost-effective manner.

The platform is cloud-friendly, scalable and secure, and allows you to build a customized intelligent search application for your website, intranet or business application. Pureinsights offers support for your search application in the form of consulting and implementation services, as well as a unique, fully managed services model we call SearchOps™.

This document provides you with an overview of PDP and its varied use cases as well as more technical information on the overall architecture, components and capabilities of the platform.

Pureinsights Discovery Platform™

Search applications today (in fact any application) revolve around the emergence of modern, cloud-based architectures. This design philosophy emphasizes loosely coupled services that are invoked when needed. The key advantage to this approach is flexible and unconstrained access to computing resources. These services can be custom services, developed in-house, or your choice of best-in-class third party services for Artificial Intelligence (AI) powered functions like Machine Learning (ML) and Natural Language Processing (NLP). Pureinsights has embraced this approach in the creation of a modern technology platform we use to build outstanding search experiences for our clients: the Pureinsights Discovery Platform™ (PDP).

PDP enables the creation of a Google-like search experience, built from best-in-class components and services. It uses a modern cloud-based architecture, and incorporates data connectors, content processing, AI services, search engines and knowledge graphs. Used together these tools provide powerful features such as: Direct Answers, Featured Snippets, 'People Also Ask' and Knowledge Panels.

These features go beyond what a typical enterprise search engine provides, enabling organizations to give their people the search experience they now expect. PDP enables developers to ingest massive amounts of content, then understand and enrich that content to find the key answers that users are likely to seek. At the same time PDP provides tools to understand the intent of a user's query so that we can deliver the most relevant, personalized and actionable search results. PDP also supports search relevance scoring and tuning.

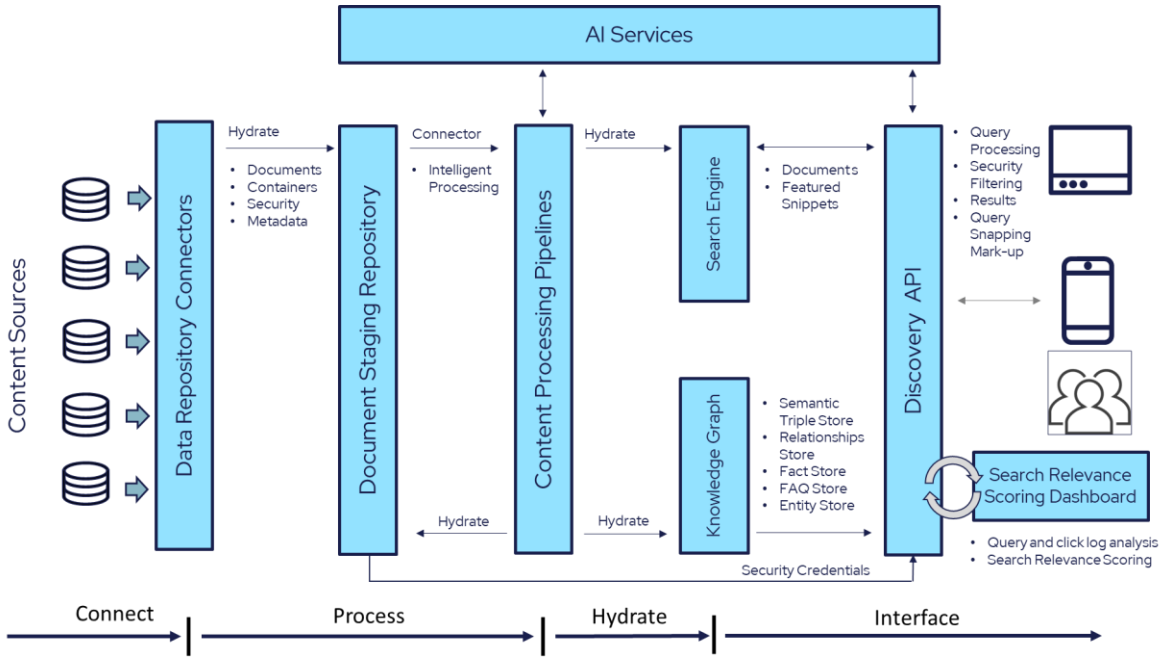
In today's loosely coupled cloud architectures, PDP is a platform that can integrate and orchestrate the best in open source or commercial technologies for any type of search application such as:

- Corporate intranets or research portals
- Customer portals for information and media publishers
- Customer support portals
- Website search
- Ecommerce search

To support these varied use cases, a platform needs to be able to:

- **Connect** to a wide range of different data sources
- **Process** and prepare (tag) these content sources for search and retrieval; reprocessing the content, as necessary.
- **Hydrate** or publish data to search engine or knowledge graph to facilitate user queries and answers
- **Interface** via a query API with different customizable applications and user interfaces

All while leveraging AI cloud services (commercial and open source) to take advantage of the most advanced natural language processing and understanding capabilities possible.



Connect

The first piece of any search project is to gain a deeper understanding of the data being searched. People expect access to content sources that will deliver relevant answers to their search queries. But this content often resides in disparate repositories such as databases, file systems, collaborative platforms, third party applications and websites. These sources might be in the cloud, in a private cloud or on-premise. PDP enables you to build connectors to any data source, then aggregate content from multiple sources. Data (raw data, documents, metadata) is ingested in a scalable and efficient manner, while honoring access controls. The connector then monitors the data source for additions, updates, and deletions and processes them as they occur.

Process, Stage, Reprocess

Analysts estimate that 80 to 90 percent of business content is unstructured and not in the best format for indexing with a search engine or ingesting into a knowledge graph. The best content is human generated (documents, presentations, social media posts, emails). These things are not usually created with search in mind. Poor quality data, especially metadata, can have a very detrimental impact on search performance. So, the next critical step in the creation of an excellent search application is to optimize that data so it can be used to answer questions effectively. PDP content processing pipelines ingest the data to clean and enrich it.

We recommend staging all the data you consume in a place where it can be analyzed and improved. PDP provides a scalable staging repository for this purpose. This staging repository has several functions. It acts as a holding area for data providing fast access when needed: for example, when publishing to a target application or re-indexing. It also makes it possible to undertake batch content processing for continuous quality improvement and testing of new content processing services, the results of which can be used to improve search engine and knowledge graph performance. Other interested applications (e.g., sales, customer, product) can also connect to the staging repository or subscribe to an event-driven model to leverage and augment data enabling an efficient 'connect once use many' approach.

Once the original data is staged, we can iteratively process it, so it is regularly cleaned, filtered, normalized, and enriched as business rules change over time. Calls to cloud-based AI services help with language identification, entity extraction, metadata extraction, tagging and classification.

Hydrate

Processed, cleansed, and enhanced data is published to an enterprise search engine and/or knowledge graph. We call this hydration. Our platform is independent of search and knowledge graph technology, and we have built hydrators to industry leading products using our toolkit. The enriched data enables advanced search features such as featured snippets, direct answers, and knowledge panels.

Interface

The goal of any search application is to serve the users' needs quickly and efficiently. PDP provides the tools necessary to build user experiences that meet the diverse needs of their communities. To fully close the loop, we need to establish intent, run the search and present results in a way that meets every user's individual needs. The platform includes a powerful search API that developers can use to create a fully personalized search experience. Sophisticated query parsing, Natural Language Processing (NLP) and other AI services are deployed to help decipher user intent. Security will be included in this API to ensure users are served only results they are allowed to see, which is crucial in the enterprise.

PDP also includes a complete React based Search User Interface that customers can deploy with minimal development effort. This UI includes Question Answering, FAQs, Extractive Answers, Knowledge Panels and all the key pieces to make your search "work like Google."

Security

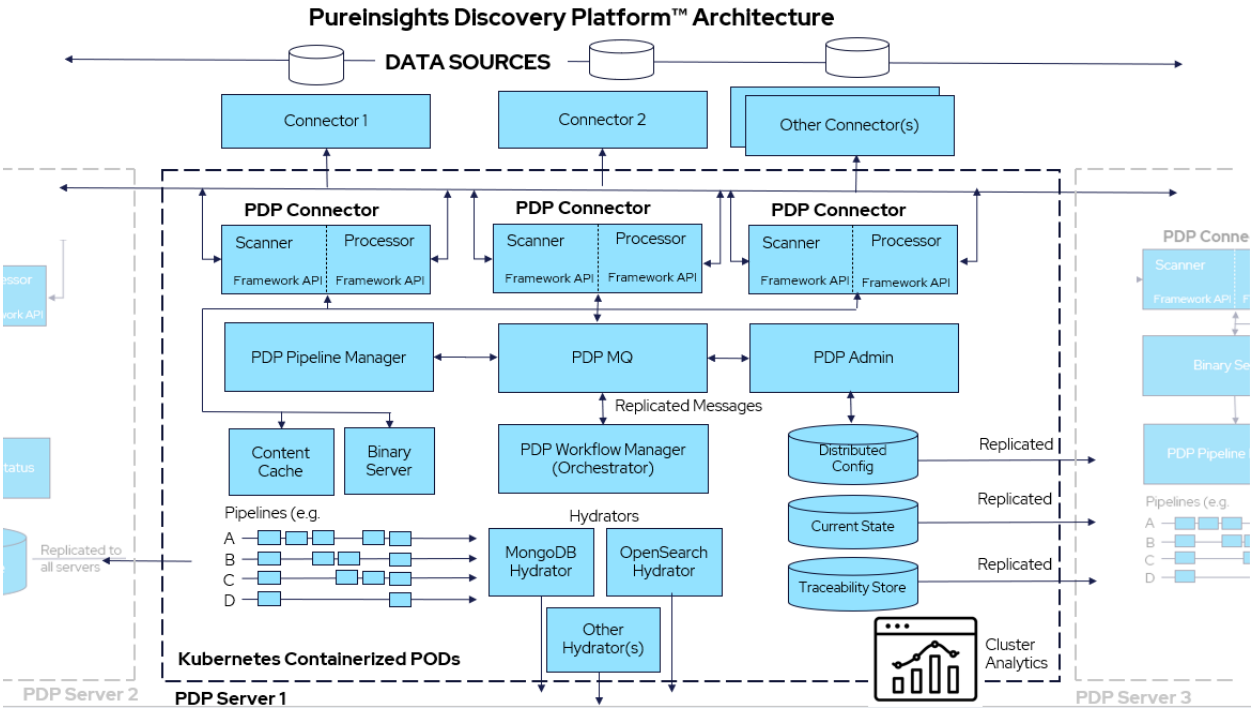
Data security and access control are critical to any application platform. PDP will have the capability to encrypt and secure data throughout its lifecycle, from ingestion, to processing, to consumption by end users or applications.

Search Relevance Scoring and Tuning

Search applications are like automobiles – they perform better with regular maintenance and tuning. PDP includes optional tools and methodologies to monitor and score the relevance of search results to end users. This is the first step in diagnosing and fixing search application issues, or to determine the impact of a change or enhancement to the search application.

Ingestion Framework

PDP was designed with the cloud in mind: easy to scale and manage, and easy to integrate with other cloud services. PDP's content processing architecture includes connectors which scan and process content from diverse sources during ingestion. PDP also includes the necessary messaging, pipeline management, orchestration, traceability, and publishing services to manage large volumes of data. Kubernetes containerized PODs leverage the latest cloud platform technologies and enables PDP to easily scale up/down as processing workloads change.



The Ingestion Framework is a custom ETL (Extract, Transform, Load) implementation that facilitates the ingestion, cleansing, normalization, augmentation and digital stitching of content from different data sources so this content can be made available through the Discovery API for use in search applications.

The Framework consists of:

- Core Components
- Admin API
- Workflow Manager and Orchestrator
- Pipeline Manager

Core Components

The core components of PDP provide the basic root functions for communications, configuration and administration of the platform. This includes:

- Binary data server – an intermediate storage to manage the processing of large files prior to uploading to the staging repository.
- Asynchronous message delivery with RabbitMQ, an open-source, high-performance message broker
- Distributed configuration store – pipeline configurations stored in a distributed data store for performance, failover and fault tolerance.
- Distributed traceability store – tracking and storing all actions in PDP for detailed visibility and analytics.

Admin API

A RESTful JSON API to allow for configuration and complete control over all PDP features. Through this API, you can create ingestion entities (pipelines, processors, seeds, etc.) and also start, stop and schedule ingestion processes.

Workflow Manager (WFM) and Orchestrator

This component acts as the "brain" of the ingestion platform. It is responsible for triggering seed executions, monitoring the progress of existing executions and cleaning up after any work that is done. WFM also optimizes the distribution of jobs throughout the cluster elastically. The workflow manager must be running at all times in active-active mode for effective content ingestion management.

Pipeline Manager

The Pipeline Manager controls the steps to follow during content processing. It works in cooperation with the Workflow Manager (WFM) to ensure ingestion executions are healthy. The Workflow Manager will assign tasks to the Pipeline Manager which get distributed across the cluster for maximum scalability. The Pipeline Manager also performs housekeeping tasks after each job / record in an execution.

Ingestion Connectors

Connectors allow PDP to retrieve data from a given source prior to content processing. Connectors first work in scan mode to detect updates and changes to the known set of records to process. This enables the framework to keep track of all the added, updated and deleted documents to be processed and uses resources efficiently when in this mode. Once scanned, records are processed. Documents or records are examined in batches by any "ingestion processor" in the pipeline associated to the content source. Failed documents or batches of documents are retried automatically to ensure maximum completeness.

PDP currently has connectors to the content sources listed below. The list will continue to grow with future versions of the software, and we expect to be able to support ingestion from all of the most popular data sources in all the most common formats. If a connector is not yet available for a given project, a custom connector can be easily developed as a service engagement using the connector framework in place in PDP.

The current connectors for PDP v1.0 represent a wide variety of sources, including:

- File System Connector – data from basic file systems or directories
- MongoDB Atlas Connector – data from an existing MongoDB Atlas database
- URL Connector – to download content from specific URLs
- Website Connector – to crawl a website or group of websites based on a starting URL
- Azure Blob Connector – data from Microsoft Azure Blog Storage
- RDBMS Connector – data from relational databases (via JDBC)
- S3 Connector – data from Amazon S3
- Udemy Connector – data from the Udemy online course director
- Elasticsearch Connector – data from any index or indexes stored in Elasticsearch

In addition, PDP has special connectors for development purposes:

- Staging Connector – records from the PDP Staging Repository (more in this document)
- Random Generator Connector – to create random data for scalability and performance testing purposes
- OpenSearch Connector – data from an existing OpenSearch index
- Elasticsearch Connector – data from an existing Elasticsearch index
- Apache Solr Connector - data from an existing Solr index

Connectors for ingesting data from existing search engine indices are useful in cases of migration from one search engine to another, or for enriching an existing index without having to recreate the index from scratch.

Ingestion Processors

After data is extracted from different content sources, a variety of services are available to transform the data obtained after ingestion. The list of processors will grow and evolve as PDP leverages newer or additional data transformation services. There is significant use of dependable open-source tools which contributes to PDP's cost effectiveness. This list also provides developers with transparency and insight into the data transformation capabilities of PDP. The most notable current ingestion / transformation processors include:

Basic Content Processors

- CSV Processor – splits a CSV file into multiple records for processing individually.
- Field Mapper – allows for simple transformations to processed data (copy fields, join fields, lowercase, uppercase…)
- HTML Processor – parses an HTML file and extracts selected subsections by class
- JSON Processor – converts a byte-array into its corresponding JSON representation
- Keyword Extraction Processor – Uses YAKE to extract keywords and key phrases from text
- Language Detector – detects the language of a text
- NLP Service Processor – uses spaCY to perform Entity Recognition, Sentiment Extraction and Dependency Parse Trees.
- OCR Processor – uses Tesseract to extract text from images
- Script Processor – allows to configure custom processing scripts. Currently supports:
    - Groovy
    - Python (through Jython)
    - JavaScript (though Rhino and Nashorn)
- Taxonomy Tagger – probabilistically tags a text based on a dictionary or taxonomy.
- Tika Processor – uses Apache Tika for content detection and extraction

Special AI / NLP Processors

- BERT Service Processor – uses BERT to vectorize chunks of text. BERT is Google's open-source, transformer-based machine learning technique for natural language processing (NLP)
- BERT Model Fine Tuning Processor – uses text received to add new vocabulary and train a BERT model; note that training new models can take considerable about of time and CPU.
- Huggingface Model runner – exposes a variety of models to perform AI and NLP tasks such as question answering (summarizing and sentiment analysis are also possible).
- Microsoft Cognitive Services – Allows PDP to pass any required document to any of the Microsoft Cognitive Services in the cloud for processing and capturing the result.

PDP can easily incorporate more basic and advanced AI-driven content processors as the state of the art improves. Google and Amazon AI/Cognitive services are on the roadmap.

Search Engine, Knowledge Graph and NoSQL DB Integration

Search engines, knowledge graphs and NoSQL databases can be integral components of any search application for two main reasons:

- They are an abstracted representation of all the underlying content and knowledge that a user might want to search on and access; and as such,
- They scale easily to support the speed and relevance of the results returned when a user query is processed

Hydrators enable PDP to write content processing results to a search index, knowledge graph, or NoSQL database. PDP supports a wide variety of hydrators for these different data storage technologies.

Hydrators

Hydrators publish processed data to different repositories or indices that are accessed to respond to search queries, or for other intermediate staging purposes.

Search Engines
- Elasticsearch Hydrator – sends the data to an Elasticsearch index
- OpenSearch Hydrator – sends the data to the OpenSearch index
- Apache SOLR Hydrator - sends the data to the Apache Solr index

Knowledge Graphs
- Neo4J Hydrator – creates the corresponding entities into Neo4j

No-SQL Databases
- MongoDB and MongoDB Atlas

Special
- Staging Hydrator – sends the data to the PDP Staging Repository (more info below)

This is not a definitive list of all the hydrators possible. Pureinsights will support other search engines, knowledge graphs and special repositories (open-source or commercial) depending on customer demand.

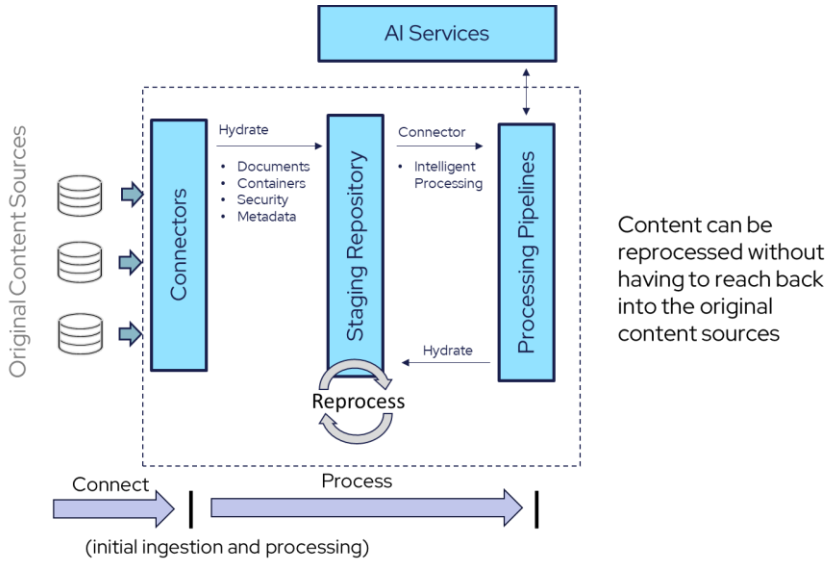
Staging Repository

The Staging Repository is an intermediate repository where content is placed after it has been extracted from a content source. This improves application performance by allowing for content reprocessing without having to reach back to the original content repository for every processing iteration.

Each Staging Repository is a storage unit, and each storage unit consists of buckets (like folders in a document system). Content is stored in the buckets, and there is a transaction log for each record stored in a bucket. The Staging Repository leverages a No-SQL database, and includes a REST API and REST client to manage, store, access and process the content stored in the repository.

Other features of the Staging Repository:

- Exposed through HTTP
- Supported No-SQL databases:
    - MongoDB and MongoDB Atlas
- Create, Read, Update and Delete (CRUD) into specific buckets
- Query/filtering
- Aggregations for deduplication
- External application subscription



Discovery API

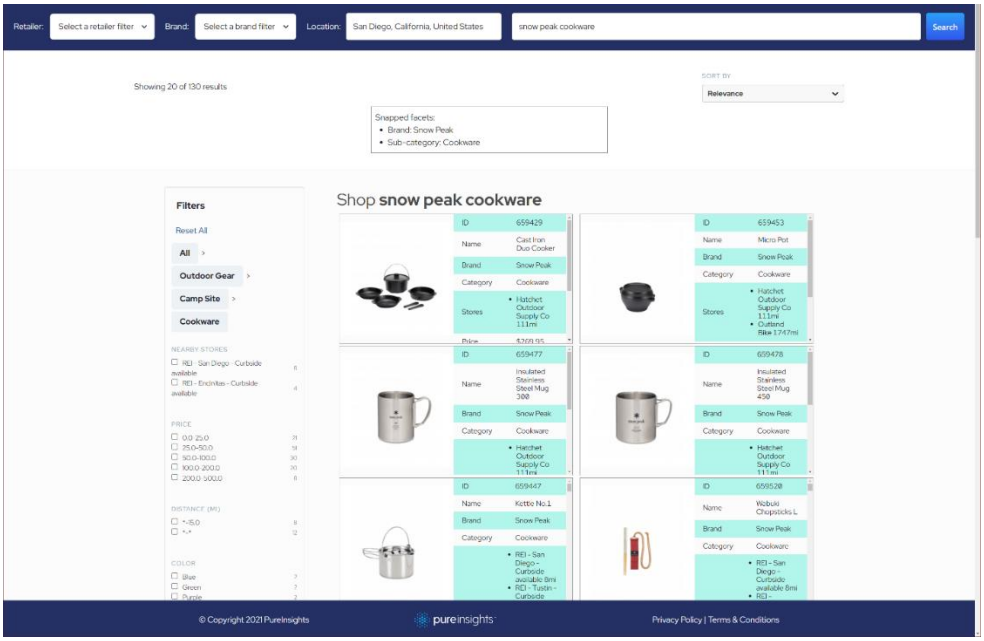
The Discovery API is designed to help an application User Interface (UI) access the data and features of underlying search engines, knowledge graphs or other special repositories. PDP has a REST API that can be used by the UI components and called directly to configure and access search engine features (like from Elasticsearch, OpenSearch or Solr).

The API supports the dynamic creation of endpoints by combining different components.

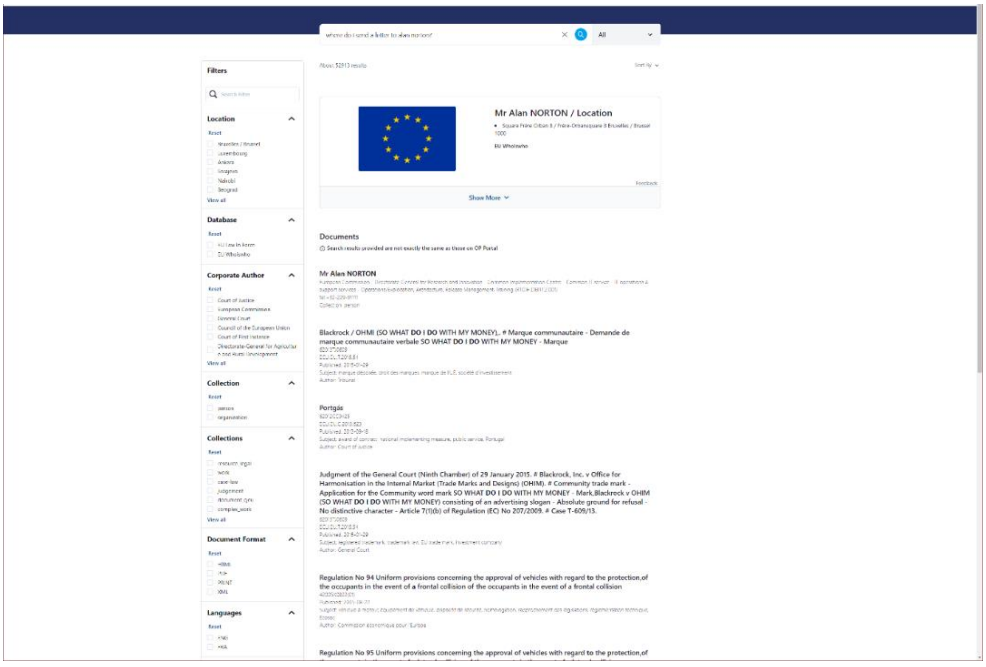
- Default Search and Autocomplete endpoints for supported search engines (e.g. Elasticsearch, OpenSearch, Solr)
- Different configuration for the components of an endpoint:
    - Sequential
    - Parallel endpoints
    - Finite-state machine
- Query fallbacks
- Supported components:
    - Search engine requests for supported search engines (e.g. Elasticsearch, OpenSearch, Solr)
    - Faceting
    - Featured snippets with DistilBERT
    - HTTP requests
    - Knowledge graph queries with Neo4j
    - Language detector
    - Parts-of-speech identification with spaCy
    - Query snapping for supported search engines (e.g. Elasticsearch, OpenSearch, Solr)
    - Query vectorization with BERT
    - Question detector
    - Request logger
    - Redirect requests
    - Script processor for custom transformation. Currently supports:
        - Groovy
        - Python (through Jython)
        - JavaScript (though Rhino and Nashorn)
    - Security filtering for Elasticsearch
    - Template-based requests

Search UI

The Pureinsights Discovery Platform™ includes components to help developers create a customized search application User Interface that provides the full, Google-like search experience for users. The UI is adaptable to different branding and functional requirements.



Example of a Search UI for E-commerce



Example of a Search UI for a Government Publications Portal

The basic functionality supported includes:

- Natural language queries (question) processing with
    - Knowledge Graph answers
    - Featured snippets / extractive answers
    - Details page / answer cards
- Traditional keyword search queries with
    - Highly relevant traditional keyword search results
    - Autocomplete
    - Pagination
    - Did you mean?
    - Query fallbacks
    - Facet Snapping
    - Results Feedback

Traditional keyword search is familiar to most search application users who may type things like "men's leather jackets." Natural language queries are when users ask the search application a full question like "How old is the moon," expecting a direct answer back.

Admin UI

The PDP product plans include a full, user-friendly Admin UI. A command line interface to admin (AdminCLI) as well as a JSON API are available in the interim.

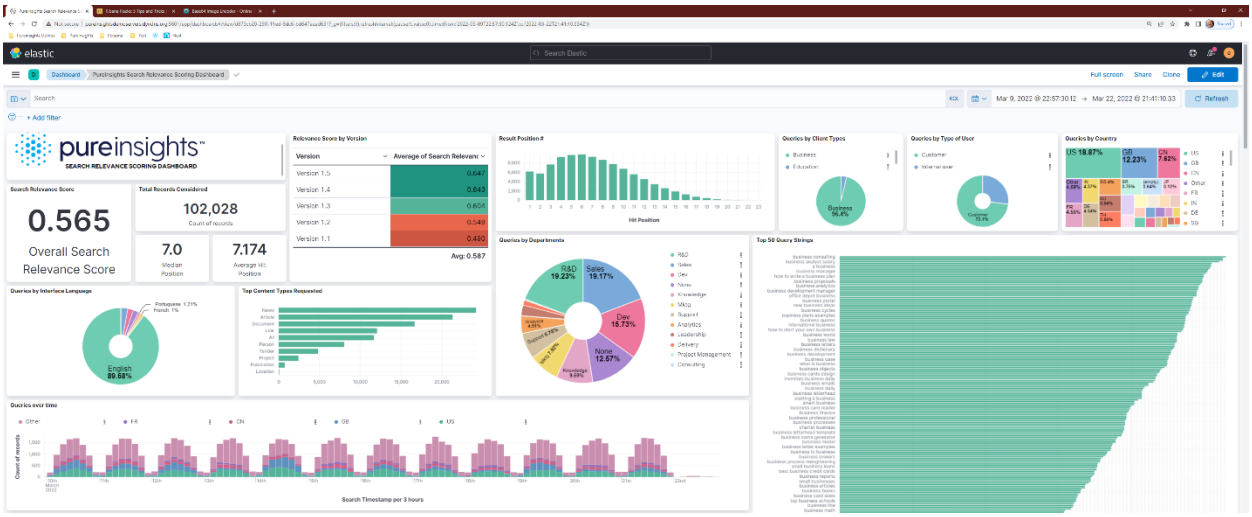
Search Relevance Scoring Dashboard

The PDP Search Relevance Scoring dashboard is a tool that provides search engine diagnostic capabilities complementary to the rest of the platform.

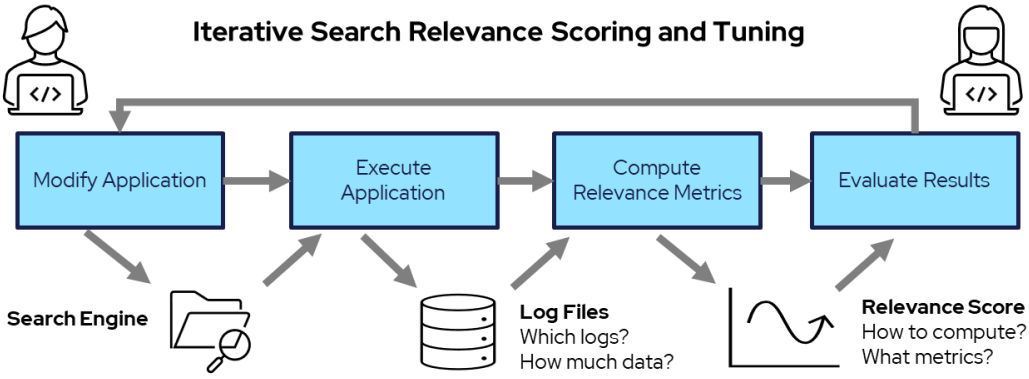
One of the goals of a search application is to deliver results that are as relevant as possible to queries submitted by users. And while search relevancy tuning requires significant expertise and a good methodology, the process starts with being able to objectively measure the quality or "score" of a search engine at any given time.

An analogy to this in medicine is the EKG or ECG (electrocardiogram). This diagnostic tool measures certain mechanical aspects of how the heart is performing and results are compared to an expected "norm." Often the EKG is the first step in diagnosing a major issue that might lead to major heart surgery. But no surgeon would even consider such drastic measures without first doing an EKG.

The PDP Search Relevance Scoring Dashboard measures the "health" of search application results. It can objectively determine when new software modifications or content changes or changing user behavior over time leads to a deterioration in the quality of search results.



The PDP Search Relevance Dashboard leverages user activity log files to analyze submitted queries and how far down the presented results users have to scroll to click on an answer, link or document they deemed most relevant. But engine scoring is a diagnostic and not a prescriptive tool. If the score is low, search engineers still have to determine the problem and introduce changes or software fixes. Regular or continuous monitoring does, however, provide an indication of whether or not the fixes have improved (or degraded) results.



More detailed information is available on how search relevance scoring works, and how it is used for relevancy tuning and continuous improvements of search results.

Use Cases

There are many ways to classify how PDP can be deployed and used to add AI and knowledge graphs to search applications. A variety of common business and functional use cases are described here to illustrate the flexibility and usefulness of the platform.

Business Use Cases

Intelligent Enterprise Search

According to a McKinsey report, employees spend 1.8 hours every day – 9.3 hours per week, or 23% of their time, on average – searching and gathering information. And in today's economy, with workers increasingly working from home, good corporate information resources have become even more critical.

But if that is the case, why are corporate intranets (traditional enterprise search) so often maligned? Why do CIOs think they have low return on investment? Why do employees always complain "our search stinks." In a chicken-or-egg situation, it is because corporate search applications are so poorly executed.

But employees rely on internet search every day. The answer to maximizing worker productivity in information search is to deliver a Google-like search experience for corporate intranets. To do this, Google leverages AI and knowledge graphs, and so should your intelligent enterprise search applications.

E-Commerce Search

E-Commerce is booming in the post-pandemic economy. And E-Commerce search represents a business use case with indisputable ROI. Research indicates that customers who use search are 2.4 times more likely to buy. They also spend 2.6 times more than other customers. Additionally, 34% of search queries are non-product queries. This could include searches on shipping options, credit options, or return policies.

The major online retailers may have addressed their customers' search needs, but the Baymard Institute declares that the state of e-commerce search is "broken", with only a handful of sites delivering a decent search experience.

One cause is that e-commerce platforms used by small and medium online retailers do not offer the search functionality necessary to deliver a good search experience. PDP can complement those platforms to cost-effectively deliver that search experience while your e-commerce platform manages the warehousing and back-office integration aspects of your website.

Customer Portals

Whether you are an online retailer, or a company that provides B2B products and services, customer portals play a key role in customer self-service strategies. According to the Service Desk Institute, a good portal can deliver business benefits such as reduced customer support costs, increased customer satisfaction, and the ability to offer round-the-clock support.

PDP can help deliver a customer portal experience that helps you realize all these benefits by delivering a search experience that can understand customers' natural language queries, and deliver the correct answer through direct answers from a knowledge graph, or extractive answers from FAQs, knowledge bases, PDF documentation, and other information sources.

Content Portals

Good search is critical for content portals provided by information and media publishers. Whether you are talking about an online subscription to the Bloomberg Finance, a free public portal like the US National archives, or a streaming service like Netflix. In this instance, the content IS the product – and the portal is the customer or user's access to the content.

PDP can be a critical component of a content platform that delivers relevant search results to natural language queries about content. PDP can also be used to power content recommendation engines, resulting in an optimal experience for content consumers.

Search & Match

Search and match applications are special search use cases where the submitted query may be an entire paragraph or document, to search for documents or content in a repository that meet certain matching criteria. Examples include matching resumes to job openings in recruiting; patent searches for potential new patents; similar research in academic repositories; or even molecular formulations in scientific databases.

In each of these (and similar) use cases, a PDP-powered search application can automate manual processes or increase the efficiencies of high-salaried knowledge workers.

Functional Use Cases

Question Answering Systems

In 2022, Google search statistics indicate that more than 21% of queries use 5 or more words – meaning users are likely typing in full questions. That figure is likely to grow significantly as people become accustomed to asking full natural language questions on search applications.

Question answering systems are viewed as the future of search. Google and Bing have trained legions of consumers to be able to type full, natural language questions in a search bar, or ask full questions from digital assistants. These online search engines use AI technologies like machine learning and natural language processing, along with knowledge graphs, do deliver the search experience user expect today.

PDP is a platform that can integrate and orchestrate the different cloud technologies available today to complement and enhance existing search applications will full Question Answering capabilities.

Knowledge Management

Knowledge management (KM) is the process by which an enterprise gathers, organizes, shares and analyzes its knowledge in a way that is easily accessible to employees. This knowledge includes technical resources, frequently asked questions, training documents and people skills.

KM is a complementary business process that can ensure the success of enterprise search or corporate intranet deployments. PDP can leverage the taxonomies, vocabularies and content management processes developed in KM to ensure that content is properly ingested, processed and indexed to ensure complete and relevant results in enterprise search applications.

Document Understanding

Document Understand helps search applications "know" what documents are about. To answer a search query like "find all construction and renovation contracts in Saudi Arabia," a search application would have to deconstruct or "understand" the query, and then submit a query to a search engine for the answer.

The result might come from a search index or knowledge graph; but to populate those databases, platforms like PDP first have to "read" or deconstruct all relevant documents (contracts) to extract key features from the document and hydrate the databases. This is the rough equivalent of "understanding" the document.

PDP leverages advanced cloud-based natural language and machine learning services like Google BERT, Amazon Comprehend, or Azure Cognitive Services to process and "understand" large-scale document repositories to support various document search applications.

Content Tagging and Processing

Users can build entire intelligent search applications around PDP – from content ingestion and processing to index and knowledge graph hydration, to comprehensive UIs. However, many information publishers and independent software / SaaS vendors (ISVs) may already have significant investments in their applications.

In this case, they can still leverage PDP to do the important job of content processing and tagging. This is the process by which content metadata is created or enriched so that search indices and knowledge graphs are hydrated with the information needed to improve search results and relevancy in the application. Even this seemingly mundane function is enhanced in PDP by the incorporation of AI and advanced NLP services.

Embedded Search Applications

Sometimes search applications do not take the form of a traditional search bar. This could include highly faceted search applications for travel reservations, GIS-based search applications, or even ride-share applications like Lyft or Uber. In these use cases, search is just one (albeit complicated) feature in a more complex application platform.

Rather than having to develop an entire search platform from scratch, independent ISVs can leverage today's API-driven cloud architectures to have just the search portion of their application powered by PDP on its own cloud infrastructure. This would allow the ISV's developers and support teams to focus on elements of the application platform that represent the core competencies of their business. PDP and the search functionality could be managed by a specialized team, or the entire search function could be delivered as a special managed service like Pureinsights' SearchOps.

Summary

Influenced by internet search, people are no longer satisfied with ranked search results from keyword queries. They want to type in full questions and get answers. They expect search to "work just like Google," with natural language queries, direct answers to factual questions and featured snippets. Traditional search is not enough. You need AI technologies and a Knowledge Graph. The Pureinsights Discovery Platform™ (PDP) brings together all the components you need to provide your users with the Google-like search experience they now expect.

Modern cloud-native architecture

The Pureinsights Discovery Platform was born in the cloud. We are building a cloud-native architecture that exploits the flexibility and resilience of cloud computing. And enables clients to run scalable, efficient, and secure search applications in modern dynamic environments such as public, private and hybrid clouds.

Data connectors

Data connectors are required to gain access to content sources such as file systems, databases and websites and feed data into PDP. PDP's data connectors ingest data in a scalable and efficient manner while honoring access controls. Plus, they monitor the data source for additions and deletions and process them as they occur.

Intelligent content processing

Poor quality data, especially metadata, can have a detrimental impact on search performance. PDP uses intelligent content processing pipelines as it ingests data to refine and optimize it for retrieval. Content is cleansed, enhanced, and normalized and can be further enriched via services for entity extraction, metadata augmentation, tagging and classification.

AI-Powered

At its core search is about understanding language and PDP takes full advantage of AI technologies such as Natural Language Processing (NLP), Machine Learning (ML) and modern Transformer Models. These AI technologies enable users to search in a way that feels natural and surface relevant results. For example, vector similarity search uses Machine Learning to provide a much more refined way to find content with subtle nuances and meanings. And featured snippets use a combination of AI technologies to extract a specific piece of text from a document that best answers a user's search request.

Integrated Knowledge Graph

A knowledge graph is a database of entities – i.e., people, places, and events – and the relationships between them. As such, they have proved enormously powerful in question-answer systems. PDP uses knowledge graph technology to provide direct answers to factual questions.

Versatile UI and search API

The goal of any search application is to serve the users' needs quickly and efficiently. PDP includes a powerful search API that developers can use to create a fully personalized 'Google-like' search experience. Sophisticated query parsing, NLP and other AI services are deployed to help understand the intent of a user's search request. Security is included in this API to ensure users are served only results they are allowed to see. PDP also includes a complete React based Search User Interface that clients can deploy with minimal development effort.

Elevate Open-Source Search

PDP has been designed to enhance traditional open-source search engines by integrating them with knowledge graph and AI technologies to provide advanced search features. You can choose to augment your existing open-source search engine with PDP or leverage the complete platform to meet the requirements of your organization.

Achieve Cost Efficiencies

PDP can help you build a better search application for your workplace, website, or support portal. It can improve the embedded search experience for your information service or software product. Combining PDP with top open-source technologies results in a cost-efficient means to achieve functionality on par with top commercial cognitive search solutions.

You can choose to augment your existing open-source search engine with PDP or leverage the complete platform to meet the requirements of your organization. Either way, Pureinsights can help you assess, design, and develop your search application and provide on-going support and maintenance.

About Pureinsights

Pureinsights has deep expertise building search applications with conventional search engines. The company helps customers go "Beyond Search", using Knowledge Graphs, Machine Learning, and Natural Language Processing to build enterprise search applications that better understand user intent and deliver answers users want. "Just make it work like Google."

For more information visit us at www.pureinsights.com or email info@pureinsights.com