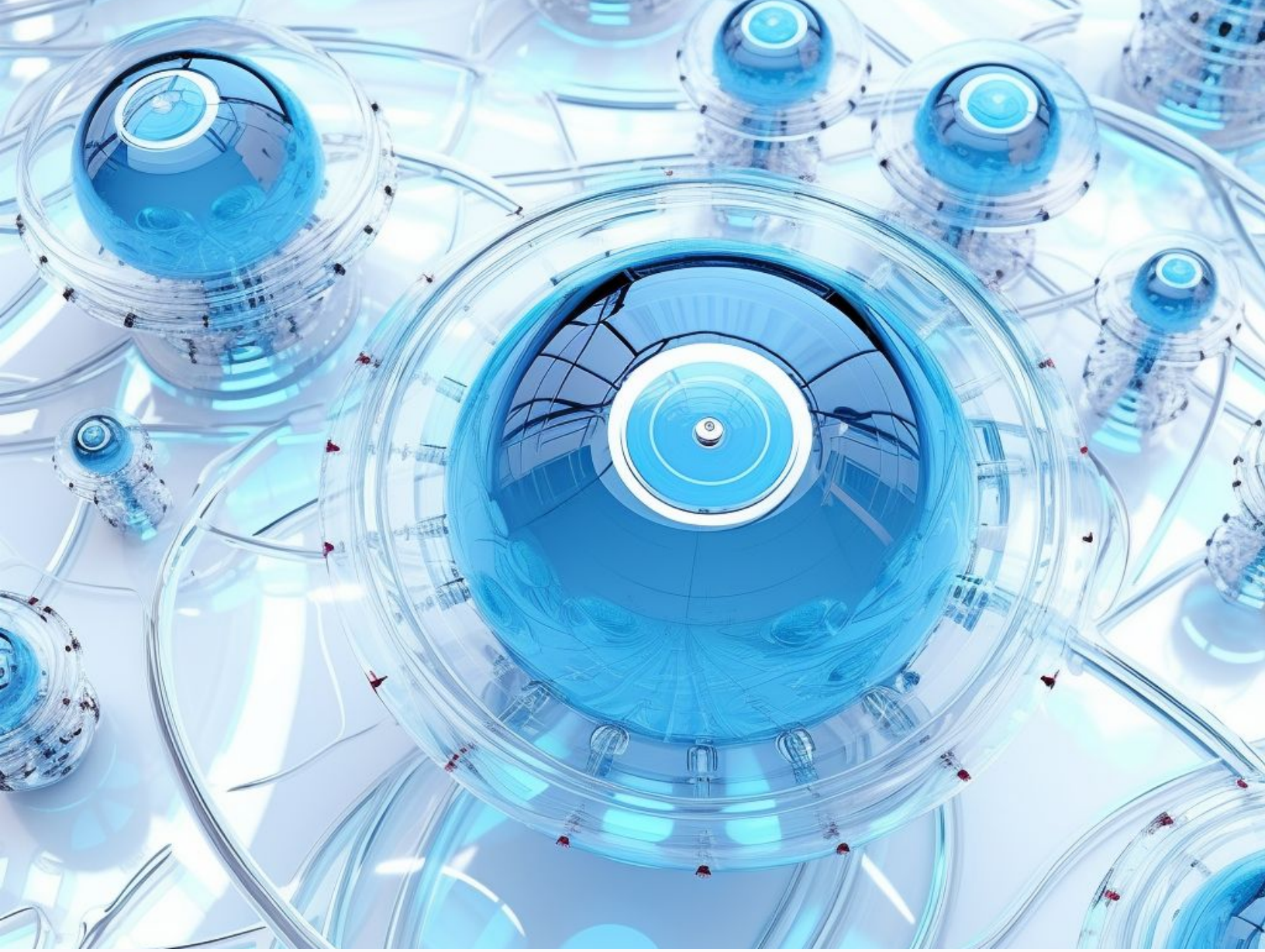

algolia

How to optimize an AI algorithm

Fine-tune pre-trained large language models for search retrieval with this look at the science behind search engine optimization.

Introduction

When it comes to AI-driven search, the best results — the most relevant ones — should always be on top. But, how does the search engine know whether a result is relevant? How are results ordered? Similarly, how can we improve relevance?

To train any machine learning model, there are essential components required, such as data, model architecture, optimizer, gradients, and objective function.

Machine learning models learn from data, so we need a dataset to train your model on. The quality and quantity of the data are critical factors that can impact the model's performance. We need to choose an appropriate model architecture that is suitable for the problem we are trying to solve.

There are various types of models, including neural networks, decision trees, and support vector machines, among others. During training, the model's parameters are updated using an optimizer, which determines the direction and magnitude of the changes to the model's parameters. In other words, the optimizer is used to improve the model. Some common optimizers include stochastic gradient descent (SGD), Adam, and Adagrad. The objective function is used to evaluate the performance of the model during training. This function represents the goal that the model is trying to achieve, and its optimization drives the learning process. Common objective functions include mean squared error (MSE), cross-entropy, and hinge loss. The gradients of the model's parameters are computed using a backpropagation technique, which involves calculating the derivative of the objective function with respect to the model's parameters.

In recent years, the field of natural language processing (NLP) has seen significant advancements with the emergence of pre-trained large language models (LLM) such as Transformers. These models have been trained on large datasets and are capable of capturing complex semantic relationships in natural language. Pre-trained models can be fine-tuned on specific tasks and domains, which can lead to improved performance on those tasks. One such task is search retrieval,

where the goal is to retrieve and rank relevant search results for a given query. Fine-tuning pre-trained LLM (Sentence Transformers) models on domain-specific data has shown promise in improving the relevance and ranking of search results. This approach can enhance the model's ability to capture domain-specific semantics and contextual information, resulting in a more effective and accurate search engine.

In this article, the general steps involved in fine-tuning pre-trained LLM models for search retrieval will be presented and the results of this approach on the publicly available dataset, namely, ESCI (Amazon) dataset, will be provided.

Data

The quality and quantity of the data can have a significant impact on the accuracy and performance of machine learning models. In fact, the quality of the data is often considered to be one of the most important factors in the success of a machine learning project. Data preparation and feature engineering are crucial steps in the machine learning process:

Data preparation

Data preparation involves collecting, cleaning, and pre-processing the data to make it suitable for analysis. This may involve removing missing values, correcting errors, and transforming the data into a format that is compatible with the chosen machine learning model.

Feature engineering

Feature engineering is the process of selecting and transforming the variables (features) in the dataset to create a set of input features that are relevant and informative for the machine learning model. This involves domain knowledge, creativity, and intuition in order to identify the most important features that will enable the model to accurately predict the target variable. Data preparation and feature engineering can be time-consuming and challenging tasks, but they are essential for building accurate and effective machine learning models. Without

high-quality data and carefully crafted features, machine learning models may not perform well, and the insights generated from the data may be inaccurate or misleading.

To fine-tune LLMs, such as BERT for domain-specific semantic and contextual information for better search retrieval, the **data used for training should ideally have relevance, diversity, quality, quantity, and labels if applicable**. The data should be relevant to the domain of interest, with examples and text passages that are representative of the language and concepts used in that domain.

For example, if you are training a model for the e-commerce domain, you should use e-commerce related texts. The data should cover a diverse range of context and perspectives within the domain, to ensure that the model can generalize well to new, unseen examples. This means including texts with different styles and types. The more data the better, as long as the quality is well maintained.

Large amounts of data can help capture a wider range of language patterns and contextual information, which can improve the model's performance. The data used for training should be of high quality, with accurate and reliable information. Data that contains errors, inconsistencies, or biases can negatively affect the model's performance. If possible, the data should be annotated with relevant labels or metadata to help the model learn more effectively.

Note that the techniques to train machine learning models aren’t sample efficient whether it is supervised learning or reinforcement learning. Therefore, it is essential to have a high quality and quantity dataset with an objective function that is the surrogate of the performance metric that the business problem is associated with. Overall, the data used to fine-tune Language Models should be carefully selected and prepared to ensure that it captures the domain-specific semantics and contextual information necessary for effective search retrieval.

The data to fine-tune LLMs to capture domain-specific semantics and contextual information for search retrieval better need to have query (string), relevance score (float), title (string), description (string), and any other feature associated with the product. An example dataset is provided on the next page. information for search

retrieval better need to have query (string), relevance score (float), title (string), description (string), and any other feature associated with the product. An example dataset is provided below.

Query	Title	Description	Relevance
10x20 ez up	American Phoenix Canopy Tent Pop Up Installation	this is a salon chair, barber chair for a haircut	Exact
10x20 ez up	ABCCANOPY Ez Up Canopy Tent with Awning	add a beautiful accent to any room with this m...	Substitute

Note: This dataset has relevance labels, such as exact, substitute, complement, and irrelevant. A relevance score of 0 is Irrelevant, 0.01 Complement, 0.1 is Substitute, and 1.0 is Exact.

Methodology

The LLMs are fine-tuned with the Contrastive loss, which is a loss function used in machine learning to train models for tasks such as text similarity or embedding. The aim of this loss function is to learn a representation of the input data in such a way that similar inputs are mapped closer to each other in the embedding space while dissimilar inputs are mapped farther apart. This loss function is particularly useful when dealing with datasets where positive and negative examples are present, and the goal is to learn a feature space that can distinguish between them. The query-title and query-description pairs form positive examples and a score (relevance score scaled from 0.0 to 1.0) is provided to guide the learning to ensure the right ranking is enforced.
The contrastive loss can be expressed mathematically as:

$$L = (1 - y) * d^2 + y * \max(0, m - d)^2$$

Where L is the loss function, y is the label indicating whether the inputs are similar (y=1.0) or dissimilar (y=0.0), d is the distance between the representations of the inputs in the embedding space, and m is a margin parameter (0.5) that specifies the minimum distance that should be maintained between the representations of negative examples. The loss function penalizes the model if the distance between positive examples is large. The contrastive loss is an effective loss function for training models for similarity and embedding tasks. It encourages the model to learn a feature space where similar inputs are mapped closer together, and dissimilar inputs are mapped farther apart.

Results

The paraphrase-multilingual-MiniLM-L12-v2 model is fine-tuned on the publicly available datasets with 5-fold cross validation. The results provided are the average performance over all folds. Each fold is trained for 10 epochs. Some additional hyperparameters used for fine-tuning are provided below:

Hyperparameters	Values	Comment
Relevance scaling	0.0-1.0	The relevance scores can be anything from human label, beta scores, or simply the ranking order. Logarithmic scaling is applied for labels from 0.0 to 1.0.
Learning rate	2e-5	The learning rate is the rate we update the parameters of the machine learning models.
Optimizer	AdamW	Adam optimizer with weight decay improves the stability of learning. The weight decay is 0.01.
Loss function	Contrastive loss	Contrastive loss function is to learn a representation of the input data in such a way that similar inputs are mapped closer to each other in the embedding space while dissimilar inputs are mapped farther apart. Cosine similarity is used within contrastive loss.

The results are provided below compared to the pre-trained paraphrase-multilingual-MiniLM-L12-v2 model. Fine-tuned model has better performance in across four different industry-standard metrics we can use to assess performance improvements — RBO with 31%, NDCG with 4%, Title cosine 32%, and Description cosine with 37%.

Table 1: Performance table comparing fine-tuned model with default pre-trained model.

MODELS	RBO	NDCG	TITLE (cosine)	DESCRIPTION (cosine)
Baseline model	0.35	0.90	0.51	0.43
Fine-tuned model	0.46	0.94	0.67	0.59

The distribution of NDCG and RBO (Figure 1) shows that not only the magnitude but also the number of positive improvements are superior to that of negatively impacted queries due to fine-tuning. Therefore, the fine-tuned model is superior to the default pre-trained baseline version in terms of ranking and relevance.

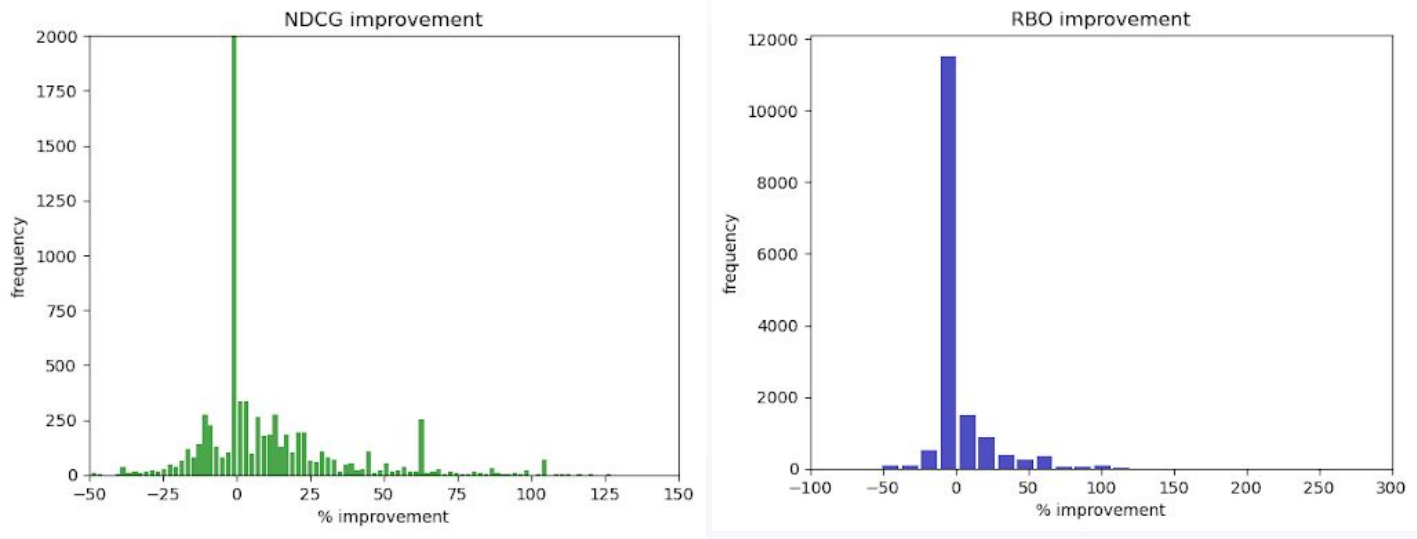

Figure 1: The distribution of RBO and NDCG improvement per query by fine-tuning the model. Note that negative improvement means the queries that are impacted negatively during fine-tuning.

Parting words

In this whitepaper, we had hoped to provide an overview of the science behind search engine optimization. We described one method for improving language understanding of large language models. However, in practice we use multiple approaches for fine-tuning including additional algorithmic adjustments and reinforcement learning. Visit Algolia's AI blog series to gain more insights.

Authored by Dr. Rasit Abay, Senior Data Scientist, Algolia

Dr. Rasit Abay is an accomplished astrodynamicist and data scientist with experience across many industries, such as logistics and last-mile, horticulture/agriculture, space, and ecommerce. With a diverse skill set encompassing NLP and Computer Vision, he is at the forefront of developing AI solutions that can comprehend and interpret textual and visual data. He is interested in improving algorithmic efficiency and data-centric AI principles for building robust and reliable AI models. He also enjoys participating in data science challenges as a competitive data scientist to solve the grand challenges of our time using data-driven techniques.

Algolia is the world's only end-to-end AI search and discovery platform. Our engineers invented a breakthrough use of AI to create exponentially better search & discovery. Our proprietary NeuralSearch tech combines vector-based natural language processing & keyword matching in a single API. Algolia powers 1.5 Trillion search requests a year or more than 30 Billion a week enabling more than 17,000 customers in 150+ countries to build blazing fast and relevant search and discovery experiences for their in-app users and/or online visitors (using any web, mobile or voice device) – by surfacing the desired content instantly and at scale.

LEARN MORE

algolia