HOW MOVEWORKS IMPROVES OVER TIME

Moveworks Intelligence Engine™

One of the main benefits of the Moveworks approach is that our customers don't need to train language models, build dialog flows, or maintain or configure the platform.

Over the last 5 years, we have pioneered a conversational AI architecture that's purpose-built for employee service. We call this underlying technology the Moveworks Intelligence Engine™. It is the driving force of our Employee Service Platform, allowing us to resolve issues instantly and automatically.

Choosing Moveworks means you're investing in a platform that automatically improves over time. The Moveworks ML team continuously invests in R&D to improve our underlying machine learning models, based on both recent and historical data across our diverse customer base.

What are the main ways Moveworks learns and improves?

Improving global models for all customers

Adapting to the unique lexicon of each customer

Learning from user feedback

Improving global models for all customers

Our team of machine learning engineers are continuously looking for ways to improve our underlying machine learning models. Improvements here benefit all Moveworks customers.

For example, Semantic Match™ is the component of our Intelligence Engine that matches questions to precise answers. A key machine learning model is our Answers Model.

Our team regularly evaluates whether certain updates to the Answer Model's parameters will improve its performance, by looking at metrics such as: the rate at which it provides good, relevant answers to employee questions, the percentage of questions for which we return a good answer, and more.

Every few weeks, our team is running an experiment to test whether we can meaningfully improve our ML models. This involves multiple steps:

1. Training the model

The process starts with a hypothesis that surfaces from analyzing thousands of queries that come through the Moveworks platform each week. For example, our ML team may have a hypothesis that adding the intent of the user's query as a feature into the Answers Model will improve its performance. The team will then use our collective data to train what we call the "candidate model" and update its parameters.

Training ML models requires a large amount of data, and is not something most organizations - even large enterprises - can do effectively. Most organizations simply do not have enough real world data inside the organization but Moveworks has mastered this process by training on over 100 million service tickets.

2. Evaluating the updated model

Once we've trained a replica model that's potentially better than the existing one, we need to evaluate it.

We use what we call a "rolling evaluation process" - we create a statistically relevant, random set of responses from our previous 8 weeks of data that have come through the Moveworks platform (several thousand queries) to test our new model. Using recent data is important, in order to account for any meaningful shifts in the real world data.

This step of evaluating results also requires a human element - judgment calls on whether answers returned by our models are relevant.

We have a team of several dozen annotators, who are constantly evaluating and labeling results from our machine learning, including potentially improved models.

3. Rolling the new model into production

If the experiment is successful, we will deploy the new model into production, replacing the old model. All customers will now benefit from the improved performance, without having to do anything.

Adapting to each customer

We have found that the world of employee service - for example, the way people describe their issues - is very similar across organizations of all sizes, industries, and geographies.

However, there are still some unique aspects for each customer that Moveworks will learn. Most commonly these are entities - such as group names, internal systems, nicknames, acronyms, and more.

For example, your IT Service Desk may be named "Juice Bar". So when your employee asks the bot "what are the juice bar hours" - you want Moveworks to serve an article or FAQ answer with your Service Desk hours of operation.

To enable this, we go through a process with each customer called entity mining. We look at all of your data, determine which terms are likely to be entities, and then go through an annotation step (human judgment) before adding it to our NLU models.

This process then improves the results for your organization only. In other words, the data we extract on entities and synonyms help improve customer-specific models, which work in conjunction with our global machine learning models.

Learning from user feedback

Once your Moveworks bot has been deployed, Moveworks will also learn from user feedback over time.

Signals here include whether a user rates a result positively or negatively (thumbs up or thumbs down), as well as what follow-on action the user takes.

For example, if the user clicks on a URL link for an answer that Moveworks provides, we also consider that a positive signal (less weight than a thumbs up). If Moveworks offers up a solution but the employee still files a ticket or requests help from a live agent, we view this as a negative signal that Moveworks didn't resolve the issue.

Over time and with enough frequency, these user interaction signals can be a more direct way of updating specific features within our machine learning models.

It's difficult to generalize how many user signals it takes to alter the result for a particular query or particular resource.

For example, when an employee asks "what are the company holidays", we may serve a knowledge article with the answer with 95% confidence. If dozens of different employees rate this response as "not helpful" - this may lower our confidence score for that particular query to 90%.

The key takeaway here is that there are many factors at play. A couple of employees, or one "bad actor" who rates the "right answer" as unhelpful, or the "wrong answer" as helpful will not have an outsized impact on the performance of your Moveworks bot.