# Named Entity Recognition | Webz.io

*Read time: 6 minutes*

---

« Back to Glossary Index

## What is Named Entity Recognition (NER)?

Named Entity Recognition (NER) is a subset of natural language processing (NLP), and NLP is a subset of machine learning (ML). NER involves identifying, classifying, and extracting named entities in text. Named entities are words or phrases that represent specific items or concepts, such as organizations, products, people, locations, dates, and times. Consider this sentence:

- *Nike releases new Air Jordan basketball shoes in the U.S., with later releases planned for Mexico and Brazil.*

A named entity recognition model would identify and extract “Nike,” “Air Jordan,” “basketball shoes,” “U.S.,” “Mexico,” and “Brazil” — along with their related category and entity labels.

### Why use NER?

NER makes data from unstructured text more accessible and actionable. By applying named entity recognition to massive volumes of text, organizations across industries can perform valuable tasks such as aggregate news,

monitor social media, identify emerging market trends, and track public perception of brands.

Some companies use named entity recognition in NLP to enhance various tasks, such as sentiment analysis. For example, a social media monitoring platform using NLP might generally analyze sentiment in the text of social media posts. However, NER can isolate entities for a more targeted analysis, identifying how users feel about specific brands or products.

NER is primarily used to extract data from unstructured text. However, NER models generally perform faster and more accurately when algorithms are trained with clean and structured data. Structured data feeds are typically well-formatted and noise-free for easy parsing and provide clearer context and boundaries for entities. These qualities are critical to platforms that require massive volumes of data for fast and accurate insights, such as media intelligence and market research intelligence platforms.

## How does named Entity Recognition work? A step-by-step breakdown

NER involves numerous processes, typically following these steps:

### Step 1 — Data collection

The first step is collecting data for your named entity recognition model. Typically, the data comes from an aggregated dataset of annotated text, with humans or machines completing the annotations. The dataset should include labeled or marked named entities, some of which may need fine-tuning. It is sometimes necessary to add newly annotated text data to address false positives in the NER model’s results.

Webz.io provides premium, noise-free structured datasets, which include datasets for training NER models. You can get a named entity recognition dataset customized through advanced filters for your specific model. Our structured datasets make named entity recognition annotation easier because they include pre-defined entity categories, such as organizations, locations, and persons.

## Step 2 — Text preprocessing

Next, the annotated text dataset goes through preprocessing, which includes:

- **Text cleaning —** Remove noise from the text, such as spam text, duplicate text, and gibberish. Correct errors in the text, such as typos and spelling mistakes.
- **Normalization —** Transform the text into a more standardized and consistent format. This process may involve removing special characters and punctuation and converting all text to lowercase.
- **Tokenization —** The text is split into individual tokens, which can represent single words, phrases, or whole sentences.
- **Part-of-speech tagging —** Each token is labeled based on its grammatical role or category, e.g., noun, verb, adjective.

Preprocessing unstructured text is a time-consuming process involving additional steps not listed above.

*Why take the time to preprocess data? The results are cleaned and structured data that is easily read by machines, making the analysis easier and faster.*

## Step 3 — Feature engineering

This step involves extracting and constructing relevant features from the preprocessed text, referred to as feature engineering. Feature engineering helps the NER model recognize and understand entities. Features commonly used in NER include:

- Gazetteer
- Word-level
- Lexical
- Character-level
- Syntactic

Note that when it comes to feature engineering, deep learning (DL) requires very little human intervention because it can automatically learn features from raw data. However, traditional machine learning requires feature engineering to have significant intervention by humans to produce good results. This article covers machine learning and deep learning techniques later.

**Step 4 — Modeling**

The first three steps lay the foundation for step 4 — building a named entity recognition model. Generally, NER modeling involves several steps:

- Feed an algorithm training data
- Save a model on what the algorithm learned
- Evaluate the model
- Fine-tune the model
- Perform model inferencing

Typically, algorithms learn from the training data how to accurately recognize and classify named entities in text. For some applications, modeling is the

final step of the NER process. However, it is often necessary for the model to undergo post-processing to refine the results.

**Step 5 — Post-processing**

For most NER applications, post-processing is the final step. After the model initially performs entity recognition and classification, it may require post-processing to refine the results or add contextual information. The steps needed in post-processing vary and can include entity linking, entity disambiguation, or boundary correction.

As you can see, NER involves many steps, some quite complicated and time-consuming. Now, let's look at some common approaches to NER.

# Techniques for Named Entity Recognition

Most organizations leverage a combination of approaches when using named entity recognition for applications. Common methods for NER include:

## Rules-based methods

Rules-based approaches involve manually defining rigid rules to identify named entities in text data. Different methods are used to create the rules, such as:

- Pattern matching
- Regular expressions
- Dictionary and gazetteer lookups

Rules-based techniques work exceptionally well for identifying and extracting entities in well-defined domains that use specific terms, such as medical and legal domains.

*While good for some use cases, rules-based methods lack the scalability, adaptability, and speed of other approaches like machine learning and deep learning.*

## Machine learning approaches

ML techniques are data-driven, employing algorithms to learn from labeled datasets, which allow the models to predict named entities. Among the most common ML models used for NER are:

- Decision trees
- Support vector machines
- Random forests

Machine learning methods work well for NER because they can handle massive volumes of data and complex data patterns.

## Deep learning approaches

DL models automatically learn and extract features (feature engineering), which NER applications can leverage to make accurate classifications and predictions. Here are some DL models and networks commonly used for NER:

- Recurrent neural network (RNN)
- Convolutional neural network (CNN)
- Long short-term memory (LSTM)

- Transformer-based models (e.g., BERT, ALBERT, XLM-RoBERTa, GPT-4)

NER applications benefit greatly from DL models because they excel at learning and understanding complex patterns and relationships in raw data.

### Statistical approaches

Statistical NER models predict named entities based on the probability of word occurrences (based on entity labels) found in the training data. Among the standard statistical models used in NER are:

- Conditional random fields (CRFs)
- Hidden Markov Models (HMM)
- Maximum Entropy Markov Models (MEMMs)

These models perform well at capturing patterns and context in text, allowing them to identify named entities accurately.

### Hybrid approaches

Many use cases for NER require extracting entities from multiple diverse sources with complex text data, leading to the need for hybrid approaches. Hybrid approaches combine two or more techniques to create more precise, fast, and flexible NER-powered applications.

*A hybrid approach can identify and extract simple and complex entities in text.*

## Use cases for Named Entity Recognition

NER has numerous practical uses across a wide range of industries, for example:

- **News aggregation —** NER plays a pivotal role in automating news aggregation processes, such as categorizing news articles and extracting key information.
- **Social media monitoring —** Many automated media intelligence platforms use NER to track specific entities in social media posts, continuously looking for brand mentions, product mentions, and emerging consumer trends.
- **Healthcare data retrieval —** Some healthcare systems leverage NER to extract medical information from electronic health records (EHRs), such as medical conditions, medications, and treatments.
- **Financial analysis —** NER is instrumental in extracting valuable insights from financial documents and online finance sources. It can extract key entities from news reports, SEC filings, and earning reports.
- **Legal document processing —** Companies in the legal sector employ NER to automatically extract relevant entities — e.g., case numbers, law firms, judge names, legal terms — from various legal documents, speeding up document review.
- **Customer support chatbots —** Named entity recognition helps chatbots identify entities customers may reference in conversations, such as product names, services offered, and order numbers. With the help of NER, chatbots can provide accurate, contextual responses.

These examples only scratch the surface of how companies can build innovative applications with NER. They can also use it to enhance existing systems.

Interested in learning how Webz.io can provide you with clean and structured data for your NER models? Talk to one of our experts today!