Named Entity Recognition

Real-time discovery of sensitive data in flight and at rest

The OpenText™ Named Entity Recognition engine is a flexible, embeddable software component that finds and extracts sensitive and high-value entities from structured, semi-structured, and unstructured data using high performance natural language processing.

Peak performance is critical for our customers, and it's not just the speed of the engine itself that matters. OpenText Named Entity Recognition supports pre-match and post-match operations to perform optimizations, such as scope control, deferred validation, and selective early termination for immediate performance gains, and componentization, extraction, and normalization for downstream efficiency.

Your product is unique. You can choose from a variety of form factors, with support for several programming languages, to find the best fit for your environment. You can choose whether to optimize for speed or for accuracy and to what degree. You can choose whether to maximize entities discovered or minimize false positives, by using the returned confidence score for each discovered entity. Combine these options to get the balance between precision, recall, performance, and convenience that is best for you and your product.

OpenText Named Entity Recognition grammars

Grammars define which entities are of interest. You can create your own or use the existing entity grammars. Gain access to OpenText's established expertise in grammar design, by using our highly targeted curated grammar modules, which provide optimized, context-aware, maintained packages for specific geographic and industry-specific entity classes to meet common use cases, such as:

- Personal data/Personally identifiable information—including fast and accurate address detection
- Protected health information
- Payment card industry entities

These curated grammars allow your product to offer extensive compliance and governance capabilities as a primary use case, straight away and without any up-front grammar development or ongoing engineering cost.

An up-to-date full list of available grammars is available in the OpenText Knowledge Discovery documentation.

Where your customers need to detect entities that are truly unique to them, our optimized grammar language can be exposed to end users for the ultimate in value-added configurability.

Better together

OpenText Named Entity Recognition is designed to work well with third-party components through the use of readily consumable APIs and architectural options. When paired with other SDKs—such as OpenText File Content Extraction for text extraction—additional cross-component functionality is unlocked, such as table extraction for improved accuracy, or pipelining for better performance.

Key benefits

Using OpenText Named Entity Recognition and our curated grammars to provide a risk reduction service to your end users who need compliance with regulations and corporate mandates.

- Defensible compliance with information provided on data sources and examples of match types.
- Quick to market—our grammars are ready to use today. There is no need to delay your product launch.
- Added differentiation with both our fully maintained grammar sets and end user customizable grammars.
- Explainable decisions through use of hand-curated grammars rather than corpus derived models.
- Wide regulation support covering many countries, languages, and entity types.
- Lower hardware costs—latency and throughput can be balanced against precision and recall to meet cost objectives at any scale.
- Reduce ongoing engineering cost as regulations and entity definitions evolve. We ensure that the grammars remain up-to-date and high-performing.
- An experienced team, we have the experience necessary to understand how to create the most efficient grammars. With a library of hundreds of grammars available, we have done this before.

Key features

Eduction

- Hybrid grammar support including dictionary lists, regular expressions, componentized grammars, and combinations.
- Contextual processing using structure and landmarks to improve recall and reduce false positives.
- Custom grammar extensions, which can be exposed to end users or become part of a value-add service.
- Compiled grammars for efficiency and IP protection—but still extendible.
- Confidence scoring to give fine control over the precision/recall curve.
- Component tagging allows specificity as to exactly which bytes match with which components in a grammar, for use in downstream processing.
- Fully streamed operation for use in pipelined architectures (such as with OpenText File Content Extraction text extraction component) for lower latency and lower memory usage.
- Early termination for lower CPU usage when risk is identified or ruled out before an entire file is processed.
- Pre-processing optimizations to accurately control scope for lower CPU usage, lower latency, and higher throughput.
- Match validation to ensure that error detection and correction codes such as checksums are valid, reducing false positives.
- Normalization of returned results such as dates, addresses, and names, for consistent downstream processing and lower risk.
- Automatic governance will redact, restrict and delete entities where they reside to ensure compliance.

Curated grammars

- Personal data/Personally identifiable information (PII) full entities covering over 46 countries and 42 languages, designed for regulations, such as GDPR (European Union), CCPA (California), KVKK (Turkey), POPI (South Africa), APP (Australia), NZPA (New Zealand), and CPPA (Canada).
- Protected health information (PHI) entities covering entity types, designed for regulations such as HIPAA (healthcare).
- Payment card industry (PCI) entities.
- Government classification, including DoD markings, controlled unclassified information (CUI), export control, and the Australian Protective Security Policy Framework.
- Sentiment analysis covering 13 languages and providing clause-level analysis, meaning that each subject within a sentence is given its own sentiment score.

System requirements by form factor

Embeddable library (Eduction SDK)

- Linux (x32 & x64), Windows (x32 & x64), macOS (x64, M1)
- C / C++, .NET, Java

Microservice (Eduction Server)

- Linux (x64), Windows (x64)
- Process-based service
- HTTP-based API
- Requires Knowledge Discovery licensing service

NiFi processor

- Linux (x64), Windows (x64)
- NiFi API
- Requires Knowledge Discovery licensing service

Visit our website to learn more.

opentext™