Resume Parsing Technology Fact Sheet

September 14, 2025

Table of Contents

1. Overview

Fastr.ai's resume parsing technology is a multi-stage machine learning pipeline built specifically for the recruiting domain. Rather than relying on a single model to perform all extraction tasks, the system decomposes the parsing challenge into distinct stages – each using techniques best suited to specific tasks. This architectural decision ensures that each component can be independently optimized, evaluated, and improved without affecting the rest of the pipeline.

The pipeline transforms raw, unstructured resume text into a richly structured candidate profile containing relationally grouped experience and education entries, normalized location data, computed experience totals, comprehensive contact information, and an exhaustive inventory of the candidate's skills. This structured output is designed to serve as the foundational data layer for all downstream processes – matching, recommendations, filtering, and analytics. Because every downstream function depends on parsing quality, the precision and completeness of the parsing stage effectively set the ceiling on overall system performance.

The system is engineered for production scale, managing tens of millions of candidates. Each stage of the pipeline has been validated through rigorous testing, and key performance metrics are disclosed in this document to support customer due diligence.

2. Pipeline Architecture

The parsing pipeline consists of five purpose-built stages, executed sequentially to progressively enrich the candidate profile:

> Document Parsing → Raw Resume Text → NER Extraction → Entity Linking → Enrichment → Skills Extraction → Certificates Extraction → Structured Profile

Each stage addresses a distinct technical challenge. The following sections describe the architecture, methodology, and rationale for each.

2.1 Document parsing

The document parsing is the first stage in the pipeline. It extracts raw texts from resume files required for downstream processing. Proper parsing ensures that all relevant elements – such as paragraphs, headings and lists – are correctly identified and preserved, along with their logical order. Maintaining the correct text sequence and layout context is essential, as errors at this stage can propagate through the pipeline, affecting downstream tasks. The system supports common layout structures including multi-column ones and a variety of input formats – including PDFs, word processing documents and plain text files.

2.2 Information Extraction (Named Entity Recognition)

The second stage employs a custom Transformer-based Named Entity Recognition (NER) model trained on a proprietary annotated dataset developed in-house. This investment in custom training data was driven by a practical constraint: publicly available NER models do not produce the label categories required for recruiting use cases. Off-the-shelf models typically recognize generic entities such as person names, organizations, and locations, but cannot distinguish between a school name and an employer.

By training on domain-specific annotated data, our NER model extracts entities with labels designed specifically for resume parsing: name, surname, and location, as well as structured education fields (degree, field of study, school name, and dates) and experience fields (job title, dates, and company name). The model achieves a mean test F-score of 0.87 across all extracted entity types. This metric is disclosed to enable direct evaluation of parsing accuracy – a level of transparency that reflects confidence in the underlying model quality.

2.3 Entity Linking

The third stage is a dedicated entity linking model that determines whether extracted entities belong together. This is the stage that transforms a flat list

of extracted entities into coherent, relationally structured data – and it is often the most underappreciated component of a parsing pipeline.

Consider a resume listing three positions held at two companies, with dates formatted inconsistently across entries. After NER extraction, the system has identified several job titles, company names, and date ranges – but these exist as disconnected labels. The entity linker's task is to correctly group each job title with its corresponding company, date range, and description, producing a coherent experience entry. The same logic applies to education: associating a degree with its field of study, institution, and graduation date.

Without a dedicated entity linking stage, parsing systems would produce fragmented profiles: job titles disconnected from their companies, dates floating without association to specific roles, and education entries that conflate multiple degrees. This stage is what separates professional-grade parsing from basic extraction.

2.4 Enrichment and Post-Processing

The fourth stage applies a stack of in-house algorithms on top of outputs from the previous stages to compute derived fields that recruiters rely on daily, but that are non-trivial to extract correctly from raw parsed data.

Total years of experience with overlap handling. Many candidates hold concurrent positions – consulting engagements alongside full-time roles, part-time work during education, or multiple freelance contracts running in parallel. Naive experience computation simply sums all date ranges, inflating the total. Our enrichment layer detects overlapping intervals and computes the true, non-duplicated total, producing an accurate experience figure that recruiters can trust for screening thresholds.

Multi-level location normalization. A single raw location string such as "San Francisco, CA" is decomposed into structured fields: region, city, country, metro area, and zip code. This normalization enables flexible geographic filtering at multiple granularity levels – from broad regional searches to precise distance-based matching – without requiring candidates to have entered their location in any particular format.

Contact information extraction. The enrichment layer identifies and structures email addresses, LinkedIn profile URLs, phone numbers, and other social media links, saving recruiters the manual effort of locating contact details across varied resume layouts.

2.5 Skills Extraction

The fifth stage performs comprehensive skills extraction using a curated taxonomy of approximately 64,000 skills combined with a context-aware filtering model. This two-step approach – broad detection followed by intelligent filtering – achieves both high recall and high precision.

Taxonomy construction. The skills database is assembled from three established labor market intelligence sources: Nesta's Open Jobs Observatory (UK labor market data), the Lightcast Open Skills Library (formerly EMSI), and the O*NET Online Database maintained by the U.S. Department of Labor (covering abilities, knowledge, skills, technology, and tools). These sources are combined through manual curation, semi-automated annotation, and custom rules to produce a unified taxonomy with surface forms mapped to preferred labels and soft/hard skill classifications.

Phrase matching and detection. Skills are initially detected in resume text using phrase matching against the taxonomy's surface forms. This approach ensures comprehensive coverage across the full 64,000-skill vocabulary, capturing both common and niche skills that keyword-based approaches would miss.

Context-aware BERT-based filtering. Phrase matching alone produces false positives when skill names are ambiguous. For example, the word "Jasmine" appears in the skills taxonomy as a JavaScript testing framework — but it is also a common first name. A naive phrase matcher would flag "Resume of Jasmine Smith" as containing the Jasmine framework skill. Our system addresses this with a BERT language model that encodes the context surrounding each potential skill detection. The model examines left and right context windows to determine whether the detected term is genuinely being used as a skill reference. In the example above, the context "Resume of ___ Smith" signals the person's name, while "Knowledge of frameworks: ___, Jest, Mocha" confirms a technology reference. This context-aware model achieves 92% overall accuracy on the disambiguation task, with a 95% precision rate on false positive rejection – ensuring that the skills inventory produced for each candidate is both comprehensive and reliable.

The result is a complete skill inventory for each candidate – not a binary presence check against a short list of requested skills. This comprehensive extraction enables rich downstream capabilities: granular multi-skill matching, skill gap analysis, skill-based career recommendations, and labor market analytics. Systems that only verify whether a candidate possesses at least one requested skill cannot support this level of analytical depth.

2.6 Certificates Extraction

The sixth stage performs comprehensive certificate, license and registration extraction using a curated taxonomy of approximately 350 certificates combined with a meta-data grouping which points which industry does the certificate relate to. This two-step approach – broad detection using taxonomy followed by fuzzy matching combined with regular expression rules – achieves a high coverage of detection.

The taxonomy handles the following types of formal requirements: a) certifications, b) registrations, c) licenses, d) official documents required to practice, etc. The result is a complete certificate inventory for each

candidate, including some information regarding the industry typically associated with that certification.

3. Structured Outputs

The parsing pipeline produces a richly structured candidate profile organized into four primary categories. Each field is immediately usable for downstream matching, filtering, and analytics without additional processing.

Category	Extracted Fields
Personal Information	First name, last name, email, phone, LinkedIn URL, social media links, and other contact information.
Location	Region, city, country, metro area, and zip code (normalized from raw location text).
Experience	Job title, company name, start date, end date, and computed years of experience – grouped into coherent entries via entity linking.
Education	Degree, field of study, school name, and dates – grouped into coherent entries via entity linking.
Skills	Complete skill inventory extracted from the full resume text, each skill verified by context-aware disambiguation against a 64,000-skill taxonomy.
Certificates	Set of certificates, licenses, official registrations extracted from the full resume text – grouped into coherent industries.

This output is not a flat list of keywords or category labels. It is a relationally structured profile in which experience entries contain their associated job titles, companies, and dates as a grouped unit; education entries contain their associated degrees, schools, and fields of study; and skills are individually identified and verified. This structural richness is what enables precise downstream matching – the system can evaluate candidates against multi-dimensional criteria (specific role + specific industry + specific skill set + specific experience level + specific location) rather than relying on simple keyword overlap.

4. Performance and Quality

Fastr.ai discloses parsing performance metrics to support customer evaluation and due diligence. The availability of these metrics reflects the maturity and confidence of the underlying models.

Pipeline Stage	Metric	Score
NER (Information Extraction)	Mean Test F-Score	0.87
NER (Information Extraction)	Entity Types Extracted	>10
Skills Disambiguation	Overall Test Accuracy	92%
Skills Disambiguation	False Positive Rejection (Precision)	0.95
Skills Disambiguation	False Positive Rejection (Recall)	0.95
Skills Disambiguation	True Skill Detection (F1)	0.73
Skills Taxonomy	Total Skills Covered	approx. 64k
Skills Taxonomy	Source Taxonomies	Nesta, Lightcast, O*NET
Certificates Taxonomy	Total Certificates Covered	approx. 350

NER F-score context. The 0.87 mean test F-score represents balanced performance across precision and recall for all entity types. This metric was computed on a held-out test set independent of training data, reflecting real-world extraction accuracy on previously unseen resumes.

Skills disambiguation context. The 92% overall accuracy is achieved on an inherently imbalanced task—most phrase matches in a resume are not genuine skill references, making false positive rejection the primary objective. The model's 0.95 precision on false positive rejection means that 95% of non-skill detections are correctly filtered out, while the 0.73 F1-score on true skill detection reflects the difficulty of confirming genuine skills in ambiguous contexts. The deliberate calibration toward high-precision false positive rejection ensures that the skill inventories presented to recruiters contain minimal noise.

5. Limitations

Transparency about system limitations is essential for appropriate deployment and expectation-setting. The following are known constraints of the current parsing pipeline.

- Entity extraction accuracy. The NER model's mean test F-score of 0.87 means that approximately 13% of entities may be missed or incorrectly extracted in a given resume. While this represents strong performance for the domain, human review remains important for high-stakes decisions. Performance may vary across languages, resume formats, and document quality.
- Skills taxonomy currency. The labor market evolves continuously, with new skills, tools, and technologies emerging regularly. The skills taxonomy requires periodic updates to maintain coverage. Between updates, very recently coined skill terms may not be detected.
- Skills disambiguation trade-offs. The context-aware filtering model's F1-score of 0.73 on true skill detection indicates that some genuine skill references may be filtered out when their surrounding context is ambiguous. This reflects a deliberate calibration toward precision (minimizing false positives), which is the preferred trade-off for downstream matching accuracy.
- Input quality dependency. Parsing quality is dependent on input quality. Poorly formatted resumes, image-based documents without OCR pre-processing, or heavily stylized layouts may produce lower-quality extraction. The system performs best on text-extractable documents with standard resume structures.
- Entity linking scope. The entity linking model operates within education and experience sections. Entities that appear outside these standard sections (e.g., in free-form summary paragraphs) may not be linked into structured entries.

6. Responsible AI

6.1 Bias Considerations in Parsing

The parsing pipeline is designed with bias prevention as a core architectural consideration. Several design decisions directly address the risk of introducing or amplifying bias at the data extraction stage.

Name extraction is isolated from scoring. The NER model extracts candidate names (first name and surname) solely for identification and deduplication purposes. Name fields are not used as inputs to any matching, scoring, or ranking process. This architectural separation ensures that name-correlated biases—which have been extensively documented in hiring research—cannot propagate through the parsing stage into downstream decisions.

Skills taxonomy sourced from neutral, established databases. The skills taxonomy is constructed from recognized labor market intelligence providers (Nesta, Lightcast, O*NET)—institutional sources that undergo their own review processes. This sourcing approach avoids the risk of ad-hoc keyword lists that may inadvertently reflect the biases of their creators or favor terminology used by specific demographic groups.

Context-aware filtering reduces demographic false positives. The BERT-based skills disambiguation model directly addresses a class of bias risk: candidates whose personal names overlap with skill terminology (e.g., “Jasmine,” “Ruby,” “Chase”) could be incorrectly attributed skills they do not possess, or conversely, could have genuine skill references misclassified as name mentions. The context-aware model resolves these ambiguities based on linguistic context rather than making assumptions based on the terms themselves.

6.2 Transparency and Explainability

The full pipeline architecture is disclosed in this document, including model types, training approaches, data sources, and performance metrics. Each parsing decision can be traced through the pipeline stages: an entity’s extraction can be attributed to the NER model, its structural grouping to the entity linker, and its enriched fields to the post-processing algorithms. This traceability supports auditability requirements and enables meaningful human oversight of automated parsing decisions.

Performance metrics are published to enable independent evaluation. The deliberate disclosure of both strengths (0.87 NER F-score, 92% skills disambiguation accuracy) and limitations (the 0.73 true skill detection F1) reflects a commitment to honest representation of system capabilities.

Appendix: Career Path Intelligence

RESEARCH & DEVELOPMENT — This capability is currently in the research and development phase and is not yet available as a production feature. It is presented here as evidence of the platform's investment roadmap in next-generation recruiting intelligence.

A natural extension of the pipeline's comprehensive skills extraction is the ability to recommend career paths. Because the parsing system produces a detailed, verified skills inventory for each candidate, it becomes possible to compare that inventory against the skill profiles associated with known occupations and suggest both typical and alternative career trajectories.

Occupation taxonomy. Career paths are grounded in the O*NET Occupation Taxonomy (with manual modifications), which provides standardized descriptions of typical duties and required skills for each occupation. The skills most representative of each career are identified using TF-IDF scoring applied to occupation duty descriptions.

Skills embeddings. Using Fastr.ai's candidate data, skills are represented as vector embeddings trained with a Word2Vec-inspired model. This places all skills in a common vector space where semantic similarity between skills can be computed—for example, recognizing that "statistical inference" and "applied statistics" are closely related, even though they are distinct taxonomy entries.

Recommendation algorithm. A custom algorithm uses the similarity between a candidate's skill embeddings and each career's skill profile, combined with a voting mechanism, to compute a final candidate-to-career similarity score. This score is used to propose both typical career paths (high similarity to the candidate's current profile) and alternative paths (moderate similarity with specific skill gaps identified).

Gap analysis. For each recommended career path, the system identifies: skills the candidate already possesses that are relevant to the career, and essential skills the candidate would need to develop. This gap analysis provides actionable guidance for career development planning and targeted upskilling recommendations.

For questions or additional technical details, contact your Fastr.ai account representative.