Automated Data Extraction from Unstructured Text Using LLMs: A Scalable Workflow for Stata Users

Loreta Isaraj
Institute for Research on Sustainable Economic Growth,
Italian National Research Council (IRCrES – CNR)

XVIII Italian Stata Users Conference
Milan, 25 September 2025

Why is Extraction Necessary?

- Over 80% of data is unstructured (Bloomberg, 2024).
- Unstructured data is growing faster than structured data.
- Text is rich in facts, numbers, and relations, but Stata (and statistics in general) require **structured variables**.
- **Without extraction:** data locked in PDFs, reports, notes.
- **With extraction:** quantitative analysis possible, automation of repetitive coding tasks, scalability to thousands of documents.

Traditional Methods in Data Extraction: Rule-based / Lexicon

- Predefined dictionaries, keyword lists, regular expressions.
- Accurate but rigid; labor-intensive; domain-specific.
- Limited adaptability across datasets and institutions.

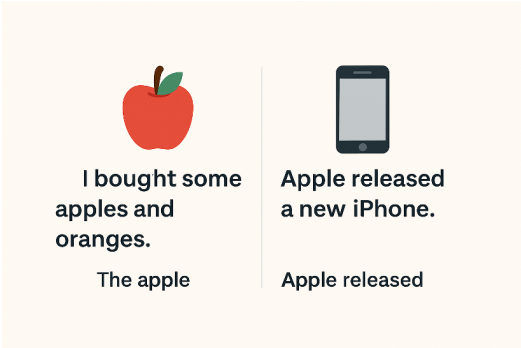
Traditional Methods: Classical Machine Learning

- Algorithms such as Conditional Random Fields (CRF), Support Vector Machines (SVMs).
- Require feature engineering and annotated training data.
- Perform better than rules but still need domain expertise.

Traditional Methods: Early Deep Learning (Domain-specific Transformers)

- BERT, RoBERTa, and domain-adapted variants (e.g., LegalBERT, FinBERT, SciBERT, ClinicalBERT).
- High accuracy when large annotated datasets are available.
- Time- and resource-intensive (annotation, retraining).

Transition to LLMs



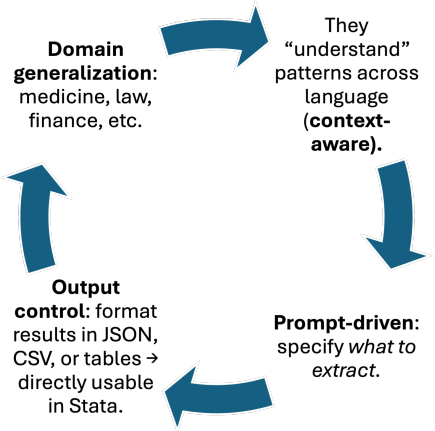


Figure: From symbolic methods to contextual understanding with LLMs.

What Are LLMs?

- AI systems trained on massive text corpora to predict and generate human-like text.
- Key idea: **next-word prediction (Autoregressive Language Models)**

$$P(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} P(x_t \mid x_{<t})$$

x_t means the word (or token) at position t.

$x_{<t}$ means all the words that came before it.

- Applications: summarization, translation, **information extraction**, coding, etc.

From Text to Numbers

Sentence: *"The patient reports pain."*

Steps

1. **Encoding:** Characters mapped to Unicode/UTF-8 numbers (e.g., "T" = 84, "h" = 104, "e" = 101)
2. **Tokenization:** [The, patient, reports, pain]
3. **Token IDs:** [101, 5023, 3841, 921]
4. **Embeddings:** Dense vectors
5. **Decoding (reverse):** Numbers back to human-readable text

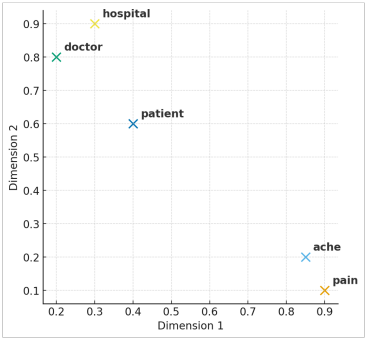


Figure: Word embeddings projected in 2D space.

Neural Architecture: Self-Attention

- The **self-attention mechanism** (Vaswani et al., 2017) is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Q = query vectors, K = key vectors, V = value vectors, d_k = key dimension.

$K^\top$ = transpose of K, flips rows/columns so queries can be compared with all keys.

- Lets each token attend to *all* others, weighting their importance.
- Enables long-range dependencies within the same sequence.

Example: "The patient reports pain"

Query ↓ / Key →	The	patient	reports	pain
The	**self**	low	low	low
patient	low	**self**	medium	**high**
reports	low	medium	**self**	medium
pain	low	**high**	medium	**self**

High = strong connection, Low = weak connection

Neural Architecture: Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W^O$$

$$\text{where head}_i = \text{Attention}(QW_i^Q,\ KW_i^K,\ VW_i^V)$$

where W_i^Q, W_i^K, W_i^V are learned projection matrices that map the input into queries, keys, and values for head i, and W^O is the output projection matrix that combines all heads back into one representation.

- Computes attention in **parallel heads**, each focusing on different relations (syntax, coreference, semantics, ...).
- Concatenating heads and projecting with W^O yields a **richer representation**.
- Improves modeling of multi-faceted context while keeping per-head dimensions small.

Multi-Head Example: "The patient reports pain"

Head	Focus	Strong Link	Interpretation
Head 1	Syntax	patient → reports	Subject–verb relation
Head 2	Semantics	patient ↔ pain	Who has the pain
Head 3	Position	reports → pain	Verb–object order

How Do LLMs Learn?

- Loss function: **cross-entropy loss** (negative log-likelihood) $\rightarrow$ maximize likelihood of training text.

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta)$$

x_t: the **true token** at position t

$x_{<t}$: the **context**, all tokens before t

θ: the **model parameters** (billions of weights updated during training)

- Predict next word, adjust parameters, minimize errors
- Variants: **Masked token prediction** (BERT), supervised fine-tuning
- Beyond cross-entropy: **Reinforcement Learning from Human Feedback (RLHF)** for alignment (e.g., GPT-4)

Ways to Extract Data with LLMs

- Conversational / Interactive (ChatGPT, Claude).
- API Integration (OpenAI, Anthropic).
- SDKs & Libraries (LangChain, LlamaIndex, Haystack).
- Fine-tuned / Domain-specific models: Train or adapt models for clinical, legal, etc.
- On-Premise / Open-Source (LLaMA, GPT-J, Falcon, Mistral).

Prompting Strategies — Direct Prompting

- **Zero-shot:** ask without examples.
  Example: Extract all medication names from "The patient was prescribed aspirin and metformin." $\Rightarrow$ [aspirin, metformin]
- **Few-shot:** provide a few labeled examples.
  Example: Text: "She takes ibuprofen." Entity: Treatment = ibuprofen.
  Now: "He has chest pain." Entity? $\Rightarrow$ Problem = chest pain
- **Chain-of-thought:** ask for reasoning steps (use with care).
  Example: "The patient was not wearing a helmet." Is this helmet use? $\Rightarrow$ mentions helmet but "not wearing" $\Rightarrow$ No

Prompting Strategies — Iterative & Feedback-based

- **Iterative refinement:** refine through multiple turns.
  Example: Extract diagnoses: "Hypertension and possible diabetes." ⇒ [hypertension].
  Follow-up: "Did you miss any?" ⇒ [hypertension, diabetes]
- **Redundant prompting:** re-ask to confirm accuracy.
  Example: "Extract blood pressure from: BP 140/90." ⇒ 140/90.
  Follow-up: "Are you sure it's blood pressure?" ⇒ Yes
- **Iterative extraction:** break task into steps (sentences → table).

Prompting Strategies — Structured

- **Schema enforcement:** force a fixed format (JSON/CSV).
  Example: "Extract medications in JSON: { "Medication": [names] }"
- **Guideline-based:** embed annotation rules.
  Example: Identify tests but *exclude* consultations as tests.
  Input: "MRI of the brain, cardiology consultation." $\Rightarrow$ [MRI of the brain]

Prompting Strategies — Advanced / Contextual

- **Error-analysis-based prompting:** refine prompts after mistakes.
  Initial: Identify tests. Output: [MRI, consultation].
  Refined: "Identify tests but do not classify consultations as tests." $\Rightarrow$ [MRI]
- **Role conditioning:** assign a role for domain tasks.
  Example: "You are a professional medical coder."
  Input: "Patient was prescribed aspirin and underwent MRI."
  Output: {Treatment: aspirin, Test: MRI}

Prompting Strategies for Data Extraction

- **Prompt quality matters:** clear, concise, structured prompts yield Stata-ready data.
- Vague or overloaded prompts lead to inconsistent, messy outputs.

Performance of LLMs in Extracting Structured Data (1)

- **Clinical/Health (SDoH in EHRs):** GPT-3.5 classified SDoH with $\kappa = 0.77$–0.85; synthetic + real data improves performance (Gabriel et al., 2024).
- **COVID-19 surveys:** GPT-4 classified infection contexts with >90% accuracy (low-entropy responses)(Bizel-Bizellot et al., 2025).
- **Emergency notes:** GPT-4 extracted helmet-use status with $\kappa = 0.74$ (low-detail prompt) up to $\kappa = 1.00$ (dictionary-based prompt); highly sensitive to prompt design (Burford et al., 2024).
- **Radiology reports:** Vicuna achieved >95% accuracy for headache/contrast detection; F1=0.97 for normal/abnormal; causal inference weaker (F1=0.85) (Le Guellec et al., 2024).

Performance of LLMs in Extracting Structured Data (2)

- **Clinical NER:** GPT-4 (5-shot) F1=0.861 (MTSamples) and 0.736 (VAERS), approaching BioClinicalBERT (0.901 / 0.802) (Yan et al., 2024).
- **Histopathology:** GPT-4 produced structured diagnoses with >90% accuracy vs human coding, enabling statistical integration (Truhn et al., 2023).
- **Materials science:** GPT-3.5/4 extracted entities/properties with >90% precision; redundancy prompting improved consistency (Polak Morgan, 2024; Dagdelen et al., 2024).
- **Systematic reviews:** GPT-4 extracted PICO elements with >85% agreement; iterative prompting increased reliability (Gartlehner et al., 2024).

Performance of LLMs in Extracting Structured Data (3)

- **Polymer science literature:** GPT-3.5 NER pipelines extracted polymer–property relations at scale (precision >90%, F1 $\approx$ 0.88) (Gupta et al., 2024).
- **Political texts:** Instruction-tuned LLaMA-2 positioned texts/actors with correlations >0.90 vs benchmarks (Le Mens Gallego, 2025).
- **Qualitative coding (sociology):** GPT-4 replicated hand-coding accuracy ($\kappa \approx 0.8$–0.9) and scaled thematic analysis (Than et al., 2025; Stuhler et al., 2025).
- **Relation extraction (general NLP):** GPT-4/3.5 outperformed supervised baselines in few-shot and unseen relation tasks (F1 $\approx$ 0.8–0.9) (Díaz-García Díaz López, 2025).

Key Parameters to Control (1) — Model

- Temperature (0–2): randomness vs determinism.
- Top-p (0–1): nucleus sampling: restricts sampling to most probable tokens (e.g., 0.9).
- Top-k: limit to top-k tokens (e.g., 50).
- Max tokens: output length cap (model dependent).
- Frequency / Presence penalties: discourage repetition / encourage novelty.

Key Parameters to Control (2) — API / Pipeline

- Batch size; parallelization for scalability.
- Rate limiting; backoff strategies.
- Error handling (e.g., retry if malformed JSON).
- Logging & versioning (track prompts, outputs, costs).

Key Parameters to Control (3) — Post-Processing

- Validation rules (e.g., dosage must be numeric).
- Normalization (units, dates, names).
- Confidence thresholds: discard low-quality results.
- Evaluation metrics (precision, recall, F1, cosine similarity).

Data Security

- Redaction (mask sensitive text); anonymization (IDs, names).
- Local vs cloud models depending on GDPR/HIPAA.
- Example: on-premis open-source (e.g., Vicuna).

Risks & Limitations of LLMs

- Hallucinations, inconsistency, privacy & compliance.
- Computational cost; bias & fairness.
- Format errors; domain knowledge gaps.

"It ain't what you do, its the extent that you do it, and that's what gets (ethically-acceptable) results."

(Dowling & Lucey, 2023)

Practical Extraction Workflow

- **Goal:** unstructured clinical notes $\rightarrow$ structured dataset $\rightarrow$ Stata analysis.
- **Takeaway:** scale from a single SOAP note to hundreds of documents, unlocking structured data for Stata pipelines.

Thank you!

Questions?

Email: loretaisaraj@cnr.it