

# AutoSpecNER: A Fine-Grained Named Entity Recognition Dataset for Vehicle Specification Extraction

Jordan Lee<sup>1,2,\*</sup>, Filippos Ventirozos<sup>1,2</sup>, Abdirahman Abdullah<sup>1</sup>, Ioanna Nteka<sup>2</sup>,  
Peter Appleby<sup>2</sup>, Matthew Shardlow<sup>1</sup>

<sup>1</sup>Department of Computing and Mathematics, Manchester Metropolitan University, UK

<sup>2</sup>Autotrader Research Group, Autotrader UK  
{f.ventirozos,m.shardlow}@mmu.ac.uk

\*Work conducted during an internship at Autotrader UK.

## Abstract

Vehicle advertisements contain rich specification information, but automotive NER resources remain limited. We introduce **AutoSpecNER**, an expert-annotated dataset for fine-grained entity recognition in vehicle listings. The dataset includes 659 advertisements from a popular car-selling website, with over 10,000 entities annotated across 15 categories, including MODEL, ENGINE\_SPEC, and BATTERY\_CAPACITY. Annotation quality was validated through inter-annotator agreement, achieving an average score of 91.5%. We benchmark rule-based extraction, fine-tuned transformer encoders, and large language models. DeBERTa achieves the best performance with a 90% micro-F1 score, outperforming the rule-based baseline (43%) and the strongest large language model (77.8%).

## 1 Introduction

The automotive industry generates vast amounts of unstructured text through online vehicle advertisements. These advertisements contain valuable specification information embedded in free-form descriptions, but extracting this structured data manually is impractical at scale. This challenge is compounded by the recent introduction of AI-generated advertisement content, which can contain hallucinations—factually incorrect information that may mislead consumers.

Named entity recognition (NER) offers a solution by automatically identifying and extracting specific information spans from text. While NER has been successfully applied to various domains including biomedical texts (Majid et al., 2024), news articles (Tjong Kim Sang and De Meulder, 2003), and social media (Derczynski et al., 2017), the automotive advertisement domain presents unique challenges:

- **Domain-specific terminology:** Technical specifications like “2.0L TDI”, “DSG transmis-

sion”, or “Santorini [black colour]” require specialised understanding.

- **Fine-grained distinctions:** Differentiating between similar concepts (e.g., exterior vs. interior colour, battery capacity vs. range).
- **Multi-word entities:** Complex specifications often span multiple tokens (e.g., “18 minutes with 350kW charger”).
- **Mixed content sources:** Advertisements include both user-generated content with typos and informal language, and AI-generated content with potential hallucinations.

This paper presents two main contributions. First, we introduce Automotive Specification NER (**AutoSpecNER**), a new, publicly available dataset<sup>1</sup> for fine-grained NER tailored to the automotive domain. The dataset contains 659 vehicle advertisements annotated with 15 entity types critical for vehicle identification and comparison.

Second, we provide a comprehensive benchmark evaluation on AutoSpecNER. We compare the performance of three diverse approaches: a rules-based system, transformer-based encoder models, and large language models (LLMs) prompted using few-shot and self-verification techniques (Wang et al., 2023). Our analysis confirms that while NER is a viable technique for this task, the choice of model has a profound impact on performance, with fine-tuned encoders demonstrating superior capabilities.

## 2 Related Work

### 2.1 Domain-Specific NER and Fine-Grained Entity Recognition

Standard NER benchmarks like CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) focus

<sup>1</sup>Available at  
[github.com/FilipposVentirozos/AutoSpecNER](https://github.com/FilipposVentirozos/AutoSpecNER)

on coarse-grained entities (person, location, organisation) inadequate for technical domains. Fine-grained entity recognition (Ling and Weld, 2012) requires distinguishing between closely related entity types—a challenge particularly acute in automotive contexts where hierarchical relationships exist between entities (e.g., “2024 Ford F-150 Limited” contains YEAR, MAKE, MODEL, and TRIM entities).

Recent work has explored product and attribute extraction (Putthividhya and Hu, 2011; Chen et al., 2023), demonstrating unique challenges in technical NER: domain-specific abbreviations, overlapping entity boundaries, and hierarchical entity relationships. Fine-grained annotation schemas have proven essential for capturing technical specifications in industrial domains (Bikaun et al., 2024), yet automotive-specific resources remain limited.

## 2.2 Automotive NER Research

The automotive domain has received minimal attention in NER research. Hu and Ma (2024) addressed NER for automotive accessories in Chinese, while recent work by Ventirozos et al. (2024) introduces the Auto-AdvER approach for English vehicle advertisements to understand the condition, historic claims and sales options offered. Recent datasets like FindVehicle (Guan et al., 2024) target vehicle retrieval with entity types including vehicle colour, brand, model, and location, but lack the granularity needed for technical specification extraction.

Park et al. (2023) introduced ADMit, combining adversarial training and multi-task learning for automotive NER for a Korean and English. Their work addresses domain adaptation between general and automotive-specific terminology but focuses on FAQ systems rather than technical specifications. Our work differs by targeting fine-grained vehicle identifiers critical for specification verification and hallucination detection.

## 2.3 Neural Approaches and Domain Adaptation

Transformer-based models have become standard for NER, with BERT (Devlin et al., 2019) and its variants achieving strong baseline performance. Domain-specific pre-training significantly improves technical entity recognition, as demonstrated in manufacturing domains where morphological patterns guide entity recognition (Li et al., 2024).

Few-shot NER methods combining knowledge graphs and contrastive learning show promise for low-resource domains (Zhang et al., 2024), addressing the scarcity of labelled automotive data. While LLMs demonstrate competitive performance through prompting (Wang et al., 2023), recent studies (Naguib et al., 2024) show that smaller, specialised models often outperform LLMs in low-resource technical domains, making them more practical for deployment in automotive applications.

## 3 The AutoSpecNER Dataset

### 3.1 Data Collection and Composition

We were provided 659 vehicle advertisements from one of the UK’s biggest online vehicle-selling websites. The dataset comprises two distinct sources:

**User-generated advertisements (350):** The dataset was a sample of an equal distribution of ads written by individual sellers (stratified across different UK counties) and dealerships (stratified across the largest dealerships in the UK). The advertisements contain natural language variation, informal descriptions, typographical errors, and inconsistent formatting.

Example: “*loyley ford focus 1.8 diesel, 2015 plate, full mot till next yr, grey metallic paint*”.

**AI-generated advertisements (309):** Created<sup>2</sup> using Google Gemini<sup>3</sup> and Meta’s LLaMA<sup>4</sup> by providing in the prompt the vehicle specifications. These are grammatically correct and well-structured but may contain hallucinations. For example in this advertisement for an Audi RSQ8, where the generated advert incorrectly refers to the vehicle as an Audi RS6:

“The Audi RS6 is a high-performance car that boasts a powerful 4.0-litre V8 engine. This petrol engine is paired with an automatic transmission...”

Other hallucinations exist which are more subtle than this such as many adverts have additional information inserted not present in the specifications, but could be true. An example of this is in the following advert in which a Volkswagen Polo Match is referred to as a Volkswagen Polo EVO Match:

<sup>2</sup>These were generated by the team of the aforementioned UK website company. These AI generated descriptions were shown to the users and the users decided whether they want to keep them or edit/delete them.

<sup>3</sup>gemini-2.0-flash-001

<sup>4</sup>llama-3.1-8b-instruct

“With only 21,029 miles on the clock, this 2021 Volkswagen Polo EVO Match is manufacturer approved...”

This dual-source approach enables investigation of how NER models handle both human errors (typos, informality) and AI errors (hallucinations, specification confusion). Crucially, each advertisement is accompanied by structured meta-data (a fact table) that lists the vehicle’s actual specifications. This is a vital characteristic, as it allows for the verification of extracted entities and forms the basis for potential error/hallucination detection systems.

### 3.2 Corpus Annotation and Inter-Annotator Agreement

To ensure the reliability and interpretability of our proposed 15-label schema, we undertook a multi-stage annotation process. The process was iterative, involving an initial schema refinement phase to produce clear guidelines<sup>5</sup>, followed by a final validation phase to confirm their efficacy.

Our initial phase involved two annotators independently labelling a pilot set of 50 advertisements, balanced between user- and AI-generated content. While this yielded a promising micro F1-score of 0.81, a qualitative analysis of the disagreements revealed systematic ambiguities. This analysis was crucial for developing a set of explicit annotation principles.

The key principles that emerged from this refinement process were:

- **Prioritise Specificity over General Description:** Labels are reserved for specific, quantifiable details, not general descriptive statements. For instance, the phrase “low CO2 emissions”, while describing an engine’s characteristic, is not annotated as *ENGINE\_SPEC* because it lacks a specific value.
- **Maintain Entity Separation for Adjacent, Distinct Items:** When multiple entities of the same type appear adjacently but refer to distinct details, they must be annotated as separate entities. For example, in the text “...TDCI 1.6L EConetic...”, each component (“TDCI”, “1.6L”, “EConetic”) is labelled as a separate *ENGINE\_SPEC* entity.
- **Disambiguate Overlapping Concepts:** Clear distinctions were established to prevent confusion. The *MAKE* label, for example, is restricted to the primary vehicle manufacturer; in ‘...a Ford Focus with a Mercedes...engine’, only “Ford” is labelled as *MAKE*. Similarly, composite names like “e-SKYACTIV G Centre-Line” are split into “e-SKYACTIV G” (*ENGINE\_SPEC*) and “Centre-Line” (*TRIM*).
- **Enforce Strict Contextual Boundaries:** Annotators were instructed to capture the full, self-contained descriptive phrase. For *BATTERY\_RANGE*, the entire phrase “maximum range of 280 miles when new” is captured, preserving the qualifying context.
- **Exclude Speculative and Ancillary Information:** The guidelines were refined to ensure that only the direct attribute of the advertised vehicle is annotated. In cases where an advert mentions other available models (e.g., “The Golf is also available as a Station Wagon (Estate)”), only the body type of the primary advertised vehicle (e.g., “Hatchback”) is to be labelled.
- **Annotate the Text, Not the Structured Specification:** Annotations should reflect the entity span as it appears in the advertisement, even when it differs from the accompanying structured specification data or appears factually incorrect. For example, if the advert describes the trim as “220d Luxury”, annotators label the span in the text rather than normalising it to the official specification value, “Luxury”. This ensures that models learn to extract the claims made in the free-text advert, which can later be compared against structured specifications to detect inconsistencies or hallucinations.

**Annotation Effort and Procedure** All annotation was carried out in Prodigy<sup>6</sup> by members of the author team rather than by crowdworkers. Each advertisement was annotated in two passes: a first full labelling pass, followed by a second pass over the same document to recover missed spans, since omission—rather than mislabelling—was the dominant error observed during the pilot. Effort scaled with specification density rather than document length: the longer but predictably structured AI-generated adverts were quicker to annotate than terse user-written ones.

<sup>5</sup>The annotation guidelines can be found at [github.com/FilipposVentirozos/AutoSpecNER](https://github.com/FilipposVentirozos/AutoSpecNER).

<sup>6</sup>[prodi.gy](https://prodi.gy)

### 3.3 Annotation Schema Overview

The final schema comprises 15 entity labels, organised into four groups and defined with representative examples in Table 1.

#### 3.3.1 Final Validation and Agreement

Following the guideline refinement, a final inter-annotator agreement study was conducted. This involved three annotators (the authors of this paper) of mixed nationalities with English as either their first or second professional language and varying levels of prior exposure to NLP annotation tasks, who labelled a new, larger random sample of 100 advertisements (50 user-generated and 50 AI-generated). Using the finalised guidelines, this validation achieved a strict-matching average micro F1-score<sup>7</sup> of **91.5%** across all three annotators (standard deviation: **3.2%**). Precision consistently exceeded recall by approximately 2 percentage points, indicating strong agreement on positive entity identification while annotators exhibited greater conservatism when entity boundaries or labels were ambiguous.

Because agreement is measured between annotators rather than against a reference, precision and recall are interchangeable under exchange of the two annotators, and we therefore report only F1. Annotation density was nonetheless consistent: the three annotators marked 765, 788 and 768 entities respectively, a spread of under 3%.

Aggregate agreement conceals variation across the schema; per-label figures are given in Table 1 and, with support and mean entity length, in Table 2. Agreement is near-ceiling for categories with canonical surface forms—MAKE (.968), YEAR (.967), MODEL (.962)—but lower for INTERIOR\_COLOUR (.639), RECHARGE\_TIME (.693) and TRANSMISSION (.732), the categories the annotators found hardest, though the small support of these labels in the agreement sample means their estimates should be read with caution. INTERIOR\_COLOUR requires a judgement of significance the guidelines could not fully operationalise (annotators divided over intermediate cases such as trim inserts and door furniture); RECHARGE\_TIME is by far the longest entity type, averaging 65.5 characters, and its disagreements concern how much qualifying context to absorb into the span; TRANSMISSION shows the same boundary problem,

<sup>7</sup>In NER in most cases Kappa is equivalent to F1, hence, we just measure F1 (Richie et al., 2022; Hripcsak and Rothschild, 2005)

as gear-system descriptions are often embedded in longer powertrain phrases.

Decomposing the disagreements (Table 3), 91.7% of annotated spans match exactly in both offsets and label, 3.9% overlap but disagree on boundaries, 3.3% are marked by one annotator of a pair only, and 1.1% assign different labels to an identical span: disagreement is chiefly about span delimitation and omission rather than category semantics. The few label confusions cluster around compound designations in which one surface string encodes several facts—“MHEV” (ENGINE\_SPEC vs. FUEL\_TYPE) or BMW’s “520d” (ENGINE\_SPEC vs. MODEL)—where a single-label schema forces a choice that we fixed by convention in the guidelines. Omissions concentrate in the two densest categories, TRIM and ENGINE\_SPEC.

| Label            | Mean F1 | Support | Len. |
|------------------|---------|---------|------|
| INTERIOR_COLOUR  | 0.639   | 16.0    | 19.3 |
| RECHARGE_TIME    | 0.693   | 17.3    | 65.5 |
| TRANSMISSION     | 0.732   | 13.3    | 7.1  |
| NO_SEATS         | 0.769   | 4.3     | 7.1  |
| EXTERIOR_COLOUR  | 0.866   | 24.7    | 15.2 |
| BODY_TYPE        | 0.867   | 20.3    | 6.7  |
| TRIM             | 0.868   | 128.7   | 8.9  |
| ENGINE_SPEC      | 0.912   | 104.3   | 4.5  |
| FUEL_TYPE        | 0.936   | 64.0    | 8.0  |
| BATTERY_CAPACITY | 0.961   | 16.7    | 6.0  |
| MODEL            | 0.962   | 177.7   | 4.7  |
| YEAR             | 0.967   | 61.3    | 4.1  |
| MAKE             | 0.968   | 114.0   | 4.9  |
| BATTERY_RANGE    | 1.000   | 10.0    | 35.0 |
| BOOT_SIZE        | 1.000   | 1.0     | 11.0 |

Table 2: Per-label inter-annotator agreement, ascending. Support is the mean number of annotated instances per annotator; Len. is the mean entity length in characters.

| Disagreement type                      | Share |
|----------------------------------------|-------|
| Exact match (offsets and label)        | 91.7% |
| Overlapping span, different boundaries | 3.9%  |
| Annotated by one annotator only        | 3.3%  |
| Identical span, different label        | 1.1%  |

Table 3: Anatomy of annotator disagreement, pooled over the three annotator pairs.

### 3.4 Dataset Statistics

The dataset contains a total of 11,117 labelled entities overall. Entity frequencies approximate a power law distribution, from MODEL (2,426 instances) to NO\_SEATS (10 instances). This reflects natural occurrence patterns - every advertisement mentions the model, but seat count is specified only when notable. One can look into Figure 1 to view the label distributions for the user and the AI advertisement generated text respectively. In total, the

dataset contains 44 different brands, 237 unique models and the vehicles range from older (first registered in 2006) to newest (this year 2025). In total, the dataset covers 44 brands and 237 unique models, with vehicles first registered between 2006 and 2025.

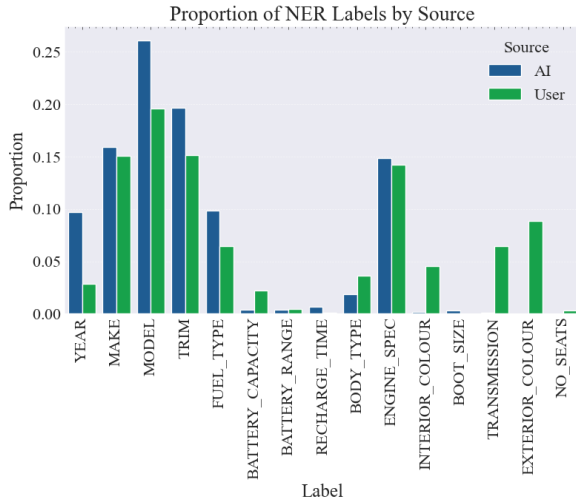


Figure 1: Proportion of labels from each source within the dataset. In dark blue are the the text advertisements generated by either Gemini or Llama and in green are the ones written by users.

## 4 Experiments

To prepare the annotated corpus for training, we performed two key steps: pre-processing the span-level annotations into a token-level format and partitioning the corpus into training (~70%), validation (~15%), and testing (~15%) sets, the distribution of the entities can be found in Table 4.

The three different ML approaches operate on different units: sub-word tokens for the encoders, whitespace tokens for the rule-based matcher, and LLM-generated free-form text. We therefore evaluate every system on a single, tokeniser-independent character-level IOB2 representation. Specifically, for each advert, the gold character-offset spans are expanded into a per-character IOB2 sequence in which every character carries exactly one tag. Within an entity span, the first character is tagged B- and each subsequent character—including whitespace within a multi-word entity—is tagged I-; every character that falls outside any entity span is tagged O (“Outside”). Each system’s predictions are mapped onto the same per-character IOB2 representation, so all systems are scored on identical units; we then reconstruct entities and report entity-level precision, recall, and

micro-F1 with seqeval (Nakayama, 2018).

| Label                      | Train       | Validation  | Test        |
|----------------------------|-------------|-------------|-------------|
| ENGINE_SPEC                | 1694        | 283         | 468         |
| MODEL                      | 1630        | 347         | 449         |
| TRIM                       | 1399        | 285         | 349         |
| MAKE                       | 838         | 172         | 219         |
| FUEL_TYPE                  | 483         | 70          | 120         |
| YEAR                       | 353         | 75          | 111         |
| EXTERIOR_COLOUR            | 297         | 58          | 125         |
| RECHARGE_TIME <sup>†</sup> | 260         | 78          | 10          |
| INTERIOR_COLOUR            | 178         | 16          | 107         |
| BODY_TYPE                  | 116         | 32          | 33          |
| TRANSMISSION <sup>†</sup>  | 114         | 9           | 16          |
| BATTERY_CAPACITY           | 86          | 44          | 32          |
| BATTERY_RANGE <sup>†</sup> | 71          | 41          | 3           |
| BOOT_SIZE <sup>†</sup>     | 16          | 8           | 12          |
| NO_SEATS <sup>†</sup>      | 4           | 0           | 6           |
| <b>Overall</b>             | <b>7539</b> | <b>1518</b> | <b>2060</b> |

Table 4: Distribution of entity counts across the final training, validation, and testing sets. <sup>†</sup> marks labels with fewer than 20 test instances: per-label scores for these are point estimates over very small samples and are not reliable for comparing models (Section 8).

### 4.1 Models

We evaluated three classes of models on the Auto-SpecNER dataset to establish a robust performance benchmark.

#### 4.1.1 Rules-Based Approach

To establish a performance baseline and explore a non-AI alternative, we developed a hybrid rules-based system for entity extraction. This approach was designed to be computationally inexpensive at inference time and serves as a benchmark against which our neural models are compared. The system combines two core methodologies: taxonomy-based matching and regular expressions.

**Methodology** Our system employs a two-pronged strategy, applying different techniques based on the nature of the target entity.

**Taxonomy-Based Matching** For eight of the core vehicle attributes (MAKE, MODEL, TRIM, FUEL\_TYPE, BODY\_TYPE, TRANSMISSION, INTERIOR\_COLOUR, and EXTERIOR\_COLOUR), we leveraged an external taxonomy provided by the company that provided the data. This taxonomy acts as a gazetteer of known values for each entity. The system iterates through the advertisement text, comparing text spans to the values in the gazetteer. We evaluated several matching strategies:

- **Strict Matching:** Requires an exact, case-insensitive string match.
- **Partial Matching:** To account for minor variations and typographical errors, we implemented partial matching using a normalised Levenshtein distance function. Predictions were made only if the similarity score exceeded a pre-determined threshold.
- **Heuristic Filtering:** To improve precision, we experimented with two disambiguation heuristics: (1) a *most-common-value* filter, which retains only the most frequently matched entity for labels expected to appear

once (e.g., MAKE); and (2) *hierarchical filtering*, which uses known make-model-trim relationships from the taxonomy to prune invalid predictions (e.g., removing a predicted trim that does not belong to the predicted model).

**Regular Expressions** For entities with highly predictable syntactic patterns that were not present in the taxonomy, we employed regular expressions. This approach was used for three specific labels: BATTERY\_CAPACITY, NUMBER\_OF\_SEATS, and YEAR. The patterns, detailed in Table 5, were designed to be robust enough to capture common formulations of these attributes.

| Label            | Pattern                          | Description                               |
|------------------|----------------------------------|-------------------------------------------|
| BATTERY_CAPACITY | <code>\b(\w+)\s+kWh\b</code>     | Matches numeric values followed by "kWh". |
| NO_SEATS         | <code>(\d+)\s*\w*seat\w*</code>  | Matches seat counts (e.g., "5 seats").    |
| YEAR             | <code>\b(?:19[0-9]\. . .)</code> | Matches four-digit years from 1900-2030.  |

Table 5: Regular expression patterns used for specific entity extraction.

#### 4.1.2 Transformer Encoders

We fine-tuned four widely-used encoder models: BERT-base-cased (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019), ModernBERT-base (Warner et al., 2025) and DeBERTa-v3-base (He et al., 2023). To perform the training and inference, we utilised the HuggingFace library (Wolf et al., 2020) for token classification.

We perform grid search over learning rates  $\{5 \times 10^{-5}, 3 \times 10^{-5}\}$ , batch sizes  $\{4, 8, 16\}$ , and weight decay values  $\{0.0, 0.01, 0.1\}$  to identify optimal hyperparameters for each encoder architecture. We train all models for a maximum of 50 epochs with early stopping patience of 3 epochs, monitoring the weighted F1 score on the validation set. The best configuration across all models uses a learning rate  $2 \times 10^{-5}$ , batch size 8, weight decay 0.01, with a warmup of 100 steps and a maximum sequence length of 512 tokens used in all experiments.

#### 4.1.3 Large Language Models

To provide a contemporary comparison to encoder-based models, we evaluated four open sourced LLMs for the NER task: Qwen3 (Yang et al., 2025) Mistral-7B (Jiang et al., 2023), LLaMA-3-8B (Grattafiori et al., 2024), Gemma-3 (Team et al., 2025b). Additionally, we evaluated two closed-sourced LLMs: Gemini-2.5 (Team et al., 2025a) and GPT-5 (Singh et al., 2026). These models were

selected as representative of smaller and resource-efficient architectures. All the specific versions with their respective parameter sizes are listed in Table 6.

**Prompt Engineering** We adopted the GPT-NER methodology (Wang et al., 2023), implementing a single-turn per-label prompting strategy, where each advertisement was processed 15 times, once for each entity type. The refined prompting approach incorporated three key components:

- **Task-specific prompts:** Each prompt focused on a single entity type with a concise task description and label definition
- **Few-shot examples:** 2 few-shot in-context learning examples were used.
- **Simplified output format:** The output would copy verbatim the example that we want labelled and the entities of interested would be marked with boundary delimiters (@@...||) rather than full NER formatting.

**Self-Verification** Following (Wang et al., 2023), a self-verification step was implemented to mitigate the high false positive rate characteristic of LLM-based NER. This secondary prompt presented the model’s initial predictions for validation, significantly improving precision whilst marginally

reducing recall. Preliminary experiments, reported in Appendix A (Appendix Table 8), demonstrated the efficacy of including this step.

Complete prompt templates are provided in Appendix B.

## 5 Results

### 5.1 Rules-based Approach

The rules-based approach utilised a combination of taxonomy-based string matching and regular expressions. Performance was highly dependent on the complexity of the entity and the matching method employed. Methods involving partial matching via Fuzzy and Levenshtein algorithms, while more accurate for variable text, were significantly slower at inference time compared to exact matching.

Table 7 summarises the best F1-scores achieved for each label. As shown, simpler, more regular entities like YEAR, MAKE, and MODEL achieved strong performance (0.77–0.79 F1). In contrast, more complex and nuanced labels such as INTERIOR\_COLOUR and EXTERIOR\_COLOUR performed poorly, with F1-scores below 0.03.

An analysis of precision and recall scores reported in Appendix C (Appendix Table 9) reveals that regex-based methods, where applicable (e.g., for YEAR), achieved high recall at the cost of precision, whereas taxonomy-based matching often resulted in higher precision but lower recall, as it failed to capture variations not present in the pre-defined lists. Overall, when evaluated across all present labels, the best-performing configuration of the rules-based approach achieved a micro-averaged F1-score of 0.43.

### 5.2 Encoder-based Models

We evaluated four transformer-based encoders: BERT, RoBERTa, ModernBERT and DeBERTa. Table 6 summarises the character-level performance of all evaluated models. Fine-tuned encoder models achieve higher performance, with micro-F1 scores ranging from 82.9% to 90.1%, compared to LLMs which achieve only 77.8% to 41.7% micro-F1.

| Model                       | Type    | Micro-F1↓ |
|-----------------------------|---------|-----------|
| microsoft/deberta-v3-base   | Encoder | 0.901     |
| FacebookAI/roberta-base     | Encoder | 0.895     |
| bert-base-cased             | Encoder | 0.873     |
| answerdotai/ModernBERT-base | Encoder | 0.829     |
| Gemini-2.5-Flash-Lite       | LLM     | 0.778     |
| GPT-5-Nano                  | LLM     | 0.775     |
| Gemma-3-27B-IT              | LLM     | 0.753     |
| Qwen3-30B-A3B-Instruct      | LLM     | 0.735     |
| Llama-3.1-8B-Instruct       | LLM     | 0.682     |
| Mistral-7B-Instruct-v0.2    | LLM     | 0.417     |

Table 6: Overall character-level performance comparison. Encoders consistently outperform LLMs with 2-shot in-context learning. The micro-F1 scores are shown in descending order.

The DeBERTa model demonstrated the best overall performance, achieving an F1 score of 90.1%. After resampling the data splits and repeating the experiments three times, the standard deviation was 2.3%, indicating relatively stable performance across runs. This strong performance may be attributable to DeBERTa’s disentangled attention mechanism, which more effectively captures positional information, a feature that is particularly relevant for the consistently structured AI-generated adverts.

Table 7 shows per-label F1 scores for all the family models. The encoders perform best on numeric entities such as YEAR and BATTERY\_CAPACITY, standardised attributes like FUEL\_TYPE and TRANSMISSION, and vehicle identifiers including MAKE and MODEL. In contrast, they struggle with more subjective attributes such as INTERIOR\_COLOUR and EXTERIOR\_COLOUR, likely due to the wide lexical variation in colour descriptions (e.g., “midnight blue”, “pearl white”, “metallic silver”).

Part of this residual error appears to be a property of the task rather than of the models. Restricting attention to the ten labels with at least 30 test instances, per-label encoder F1 correlates with the per-label inter-annotator agreement of Appendix Table 1 (Spearman  $\rho = 0.81$ ,  $p = 0.005$ ). The clearest case is INTERIOR\_COLOUR, where the best encoder scores .623 against a human agreement of .639: the model is close to the annotation ceiling, and its low score partly reflects a category whose extent annotators cannot themselves fix consistently. By contrast, EXTERIOR\_COLOUR (.725 against an agreement of .866) and BODY\_TYPE (.721 against .867) retain a genuine gap and are more promising targets for improvement.

Our analysis of the training data size indicated

that the dataset was sufficient for the task, as model performance (measured by evaluation loss) began to plateau after approximately 300–350 training samples. Further details and the corresponding

graph are available in Appendix D. The one exception is the NO\_SEATS which was scarcely reported in the text descriptions.

| Entity Type      | Best Encoder       | Encoder F1   | Best LLM              | LLM F1       | Rules F1 |
|------------------|--------------------|--------------|-----------------------|--------------|----------|
| MAKE             | RoBERTa            | <b>0.938</b> | GPT-5-Nano            | 0.825        | 0.773    |
| MODEL            | RoBERTa            | <b>0.950</b> | Gemini-2.5-Flash-Lite | 0.878        | 0.789    |
| TRIM             | DeBERTa-v3         | 0.870        | Gemini-2.5-Flash-Lite | <b>0.715</b> | 0.482    |
| YEAR             | DeBERTa-v3         | <b>0.982</b> | Gemini/Gemma          | 0.948        | 0.773    |
| BODY_TYPE        | DeBERTa-v3         | 0.721        | GPT-5-Nano            | <b>0.769</b> | 0.531    |
| FUEL_TYPE        | BERT-cased         | <b>0.957</b> | Gemini-2.5-Flash-Lite | 0.773        | 0.574    |
| TRANSMISSION     | DeBERTa-v3         | <b>0.938</b> | GPT-5-Nano            | 0.476        | 0.255    |
| ENGINE_SPEC      | DeBERTa-v3         | <b>0.933</b> | Gemini-2.5-Flash-Lite | 0.907        | 0.279    |
| EXTERIOR_COLOUR  | DeBERTa-v3         | 0.725        | GPT-5-Nano            | <b>0.763</b> | 0.024    |
| INTERIOR_COLOUR  | RoBERTa            | <b>0.623</b> | Gemini/GPT            | 0.270        | 0.007    |
| NO_SEATS         | N/A                | 0.000        | Gemma-3-27B-IT        | <b>0.750</b> | 0.258    |
| BOOT_SIZE        | DeBERTa-v3         | 0.909        | Gemini-2.5-Flash-Lite | <b>1.000</b> | —        |
| BATTERY_CAPACITY | DeBERTa-v3         | <b>1.000</b> | Gemini/GPT            | 0.714        | 0.462    |
| BATTERY_RANGE    | DeBERTa-v3/RoBERTa | 0.800        | Gemini-2.5-Flash-Lite | <b>1.000</b> | —        |
| RECHARGE_TIME    | RoBERTa            | <b>1.000</b> | N/A                   | 0.000        | —        |

Table 7: Per-label F1 scores comparing encoder, LLM, and rules-based performance. In bold are the top performers for each label. Some model names are shortened for illustration purposes but are the same version with the ones presented in Table 6.

### 5.3 LLM-based Models

The decoder models, which are much larger in parameter size—many times more than 100 times larger—than their encoder counterparts excelled in various labels, especially in those with less support, comparatively to the encoders.

Gemini-2.5 scored the highest amongst the LLMs but still second to the encoders. Its predictions across the test set reveal heterogeneous performance across entity types: strong ( $F1 > 75\%$ ) on YEAR, MAKE, and MODEL; moderate ( $F1$  50–75%) on ENGINE\_SPEC, EXTERIOR\_COLOUR, TRIM, BATTERY\_CAPACITY, and BODY\_TYPE; and poor ( $F1 < 50\%$ ) on INTERIOR\_COLOUR, TRANSMISSION, and notably RECHARGE\_TIME (0% F1 despite 10 test instances).

LLMs exhibit threetwo primary failure modes. First, *boundary detection errors*: extracting “300 miles” instead of “approximately 300 miles” (BATTERY\_RANGE), or “408” instead of “408 horsepower” (ENGINE\_SPEC). Second, *variant normalisation failures*: achieving high recall on simple FUEL\_TYPE values (“petrol”, “diesel”, “electric”) while missing hybrid variants (“Petrol Hybrid”, “Petrol Plug-in Hybrid”, “Diesel Hybrid”).

We identify three primary factors contributing to these LLM failures. First, our 2-shot learning approach may not be indicative enough for

the LLMs to perform inference. Second, vehicle specifications involve domain-specific terminology (“kWh”, “bhp”, “T-GDI”, “BiTurbo”) and complex entity boundaries (e.g., “2.0L Turbocharged Inline-4” as ENGINE\_SPEC entities), for which pre-trained LLMs lack adequate exposure during training, unlike general entities such as persons or organisations. Third, the severe lexical variation within certain entity types (e.g., INTERIOR\_COLOUR: “Bengal red Nappa leather”, “Light Oyster and Ebony”, “Full Black Valcona Leather”) exceeds what can be captured in 2-shot demonstrations, while more standardised types (YEAR, MAKE) benefit from prior knowledge in the pre-training corpus.

Albeit, That said, the LLMs proved resourceful, achieving the highest F1 score amongst the other approaches, when the support was quite low in the labels NO\_SEATS, BATTERY\_RANGE and BOOT\_SIZE.

### 5.4 Computational Resources

All encoder-based transformer models were fine-tuned on an Apple M4 MacBook. Open-source LLM inference was performed using vLLM (Kwon et al., 2023) on NVIDIA RTX PRO 6000 (96GB VRAM) or NVIDIA H200 SXM (141GB VRAM) GPUs, depending on model size. Closed-source models were accessed through their respective provider APIs. To ensure deterministic outputs, we set the temperature to 0.0 where supported and used greedy decoding for all generations. Additional

hardware and runtime details are provided in Appendix E.

## 6 Discussion

The current investigation has implications for deploying NER systems in production automotive applications. When the goal is extracting entity types and populating structured vehicle databases, fine-tuned encoder models—despite their training requirement—remain the best solution overall, with DeBERTa leading when the supporting examples are sufficient, followed by the LLMs and last the rule-based approaches.

We expect the three model families to transfer very differently to advertisements from other platforms, countries or writing styles. The rule-based system is the least portable by construction: its taxonomy-matching component is a gazetteer of makes, models and trims supplied by a single UK marketplace, so it would require a new taxonomy for every market it is deployed in, and its regular expressions assume UK conventions such as four-digit registration years and metric capacities. The encoders should degrade more gracefully, since most of what they exploit is local morphology and context rather than a closed vocabulary, but they will still encounter genuinely unseen material: manufacturer-specific colour names (“Santorini Black”, “Mythos Black”) that behave lexically like arbitrary proper nouns, market-specific abbreviations, and the right-hand-drive UK model line-ups that dominate our MODEL and TRIM distributions. Non-UK listings would additionally introduce vocabulary absent here altogether, and non-English listings fall outside the scope of these models entirely.

Looking beyond immediate deployment constraints and efficiency considerations, several avenues could improve LLM performance on this task. Increasing the few-shot to more examples per entity type could improve prediction accuracy. Similarly, retrieval-augmented prompting could dynamically retrieve relevant examples that could boost the score as well. Finally, models further pre-trained or post-trained on automotive corpora may better handle specialised terminology and entity patterns.

The high accuracy of the fine-tuned encoder models enables several valuable downstream applications. While this work focused on advertisements, the models could be evaluated for tracking

vehicles from unstructured social media posts to monitor consumer trends, popular features, and brand perception. Additionally, the process of extracting entities can be used to detect hallucinations in AI-generated content by verifying whether a model that generated the content included additional or different specifications and whether it adhered to the structured vehicle specification data given, thereby enhancing platform integrity. It was notable how on average we could spot an around 20% increase in F1 score when detecting entities in the AI generated text alone. The fine-tuned encoders could prove resourceful at tracking these mentions from AI generated text, raising flags when necessary.

## 7 Conclusions

In this paper, we addressed the high-impact challenge of extracting fine-grained vehicle specifications from the text descriptions of car advertisements. Our dataset and annotation schema **AutoSpecNER** comprises 659 advertisements—written by real users and AI-generated proprietary data acquired from the UK’s biggest online vehicle-selling website—labelled with 15 specification types ranging from MAKE and MODEL to RECHARGE\_TIME, with the schema validated at a commendable 91.5% inter-annotator F1 score, though agreement ranges from .968 for MAKE to .639 for INTERIOR\_COLOUR. Benchmarking rule-based, fine-tuned encoder, and few-shot decoder approaches, we found the fine-tuned encoders exceptionally well-suited to this task, in line with the findings of [Naguib et al. \(2024\)](#), with DeBERTa achieving a top micro F1-score of 0.901, followed by the decoders (led by Gemini-2.5) and last the rule-based approaches; the LLMs proved particularly resourceful for low-support labels scarcely reported in the text. Finally, this dataset can support market trend analysis and, crucially, the detection of factual hallucinations in AI-generated content.

## 8 Limitations

The dataset was drawn exclusively from a proprietary database maintained by a single UK-based company. Although substantial, it is unlikely to capture the full diversity of linguistic usage, document structures, and formatting conventions observed across different countries, platforms, and automotive retailers. In addition, the corpus is UK-specific and monolingual (English), preventing as-

assessment of cross-lingual performance. Improving generalisability would require a more heterogeneous, multi-source, and multilingual dataset.

**Low-support categories.** Five labels have fewer than 20 test instances—BATTERY\_RANGE (3), NO\_SEATS (6), RECHARGE\_TIME (10), BOOT\_SIZE (12) and TRANSMISSION (16), marked in Table 4—so per-label scores on these are point estimates over very small samples, and differences between models on them are not interpretable. We therefore draw conclusions about model ranking only from the micro-averaged figures. Resampling the splits and repeating the experiments three times gave a standard deviation of 2.3% in overall micro-F1, comparable to the gaps among the encoders themselves, so we do not claim that the ordering within the encoder family is significant.

**Two-shot LLM configuration.** Our LLM results characterise one prompting configuration rather than the general capability of these models. The shot count was fixed at two after a small pilot with two open-weight models using no more than ten demonstrations; this is far too small a fraction of the dataset to establish whether larger demonstration budgets would help. The demonstrations themselves are drawn from this corpus, so the LLM scores benefit from in-domain exemplars that a cold-start deployment would not have. We did not evaluate larger demonstration budgets for the closed-source models, retrieval-augmented exemplar selection, or fine-tuning; the encoder-LLM gap reported here therefore compares fine-tuned encoders with few-shot prompted LLMs specifically, and is not an upper bound on decoder-based approaches.

## 9 Ethical Considerations

The development of the **AutoSpecNER** dataset and its associated models has been guided by key ethical considerations. The dataset was constructed from publicly accessible advertisements available on a major commercial platform. A manual review of the user-generated content was conducted, which confirmed the absence of Personally Identifiable Information (PII) in the source data. While the data’s public availability and freedom from PII mitigate privacy concerns, its origin introduces a risk of socio-economic bias. The model may exhibit performance disparities across content from different seller types, and if the system is more accurate

on professionally formatted advertisements than on informal text from private sellers, its application could inadvertently create an unfair marketplace.

Consideration must also be given to the potential for misuse and downstream harms. The same technology that extracts specifications can be repurposed for malicious ends. Models trained on this data could be exploited to generate convincing but fraudulent vehicle advertisements, either for individual scams or at scale to manipulate market perceptions of a vehicle’s value and availability. In a deployed application, over-reliance on the system’s output could lead to automation bias, where users and platform administrators place undue trust in the automated extractions. This could cause model errors to be systematically propagated, leading consumers to make purchasing decisions based on incorrect information.

Finally, we address the environmental impact of this research. Our experiments involved training and evaluating multiple large-scale models, which consumed significant computational resources and energy. However, our findings make a positive contribution in this area by demonstrating that smaller, fine-tuned encoder models significantly outperform their much larger and more resource-intensive LLM counterparts for this task. This result advocates for a more sustainable approach to deploying NLP in production environments, showing that for specialised industrial tasks, more energy-efficient models can also be the most effective.

## Acknowledgements

We thank Autotrader UK for providing access to the underlying advertisement data and for funding the Sponsored Masters Programme at Manchester Metropolitan University under which this work was conducted. We also thank the anonymous reviewers for their constructive and encouraging comments.

## References

- Tyler K. Bikaun, Tim French, Michael Stewart, Wei Liu, and Melinda Hodkiewicz. 2024. [MaintIE: A fine-grained annotation schema and benchmark for information extraction from maintenance short texts](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10939–10951.
- Wei-Te Chen, Keiji Shinzato, Naoki Yoshinaga, and Yandi Xia. 2023. [Does named entity recognition](#)

- truly not scale up to real-world product attribute extraction? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 152–159, Singapore. Association for Computational Linguistics.
- Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. [Results of the WNUT2017 shared task on novel and emerging entity recognition](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Runwei Guan, Ka Lok Man, Feifan Chen, Shanliang Yao, Rongsheng Hu, Xiaohui Zhu, Jeremy Smith, Eng Gee Lim, and Yutao Yue. 2024. [Findvehicle and vehiclefinder: a ner dataset for natural language-based vehicle retrieval and a keyword-based cross-modal vehicle retrieval system](#). *Multimedia Tools and Applications*, 83(8):24841–24874.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *Preprint*, arXiv:2111.09543.
- George Hripcsak and Adam S. Rothschild. 2005. [Agreement, the f-measure, and reliability in information retrieval](#). *Journal of the American Medical Informatics Association*, 12(3):296–298. Epub 2005 Jan 31.
- Songhua Hu and Ruhui Ma. 2024. [Named entity recognition of automotive parts based on roberta-crf model](#). In *2024 4th International Conference on Neural Networks, Information and Communication Engineering (NNICE)*, pages 604–612.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP ’23*, page 611–626, New York, NY, USA. Association for Computing Machinery.
- Ruiting Li, Peiyan Wang, Libang Wang, Danqingxin Yang, and Dongfeng Cai. 2024. [A corpus and method for Chinese named entity recognition in manufacturing](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 264–272.
- Xiao Ling and Daniel S. Weld. 2012. [Fine-grained entity recognition](#). In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI’12*, page 94–100. AAAI Press.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *Preprint*, arXiv:1907.11692.
- Idris Majid, Vasudha Mishra, Raja Ravindranath, and Sophia Y. Wang. 2024. Evaluating the performance of large language models for named entity recognition in ophthalmology clinical free-text notes. *Journal of the American Medical Informatics Association*. Open Access.
- Marco Naguib, Xavier Tannier, and Aur  lie N  v  ol. 2024. [Few-shot clinical entity recognition in English, French and Spanish: masked language models outperform generative model prompting](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6829–6852, Miami, Florida, USA. Association for Computational Linguistics.
- Hiroki Nakayama. 2018. [sequeval: A python framework for sequence labeling evaluation](#). Software available from <https://github.com/chakki-works/sequeval>.
- Cheoneum Park, Seohyeong Jeong, and Juae Kim. 2023. [Admit: Improving NER in automotive domain with domain adversarial training and multi-task learning](#). *Expert Syst. Appl.*, 225(C).
- Duangmanee Putthividhya and Junling Hu. 2011. [Bootstrapped named entity recognition for product attribute extraction](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567.
- Russell Richie, Sachin Grover, and Fuchiang (Rich) Tsui. 2022. [Inter-annotator agreement is not the ceiling of machine learning performance: Evidence from a comprehensive set of simulations](#). In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 275–284, Dublin, Ireland. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025a. [Gemini: A family of highly capable multimodal models](#). *Preprint*, arXiv:2312.11805.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025b. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Filippos Ventirozos, Ioanna Nteka, Tania Nandy, Jozef Baca, Peter Appleby, and Matthew Shardlow. 2024. [Shifting NER into high gear: The Auto-AdvER approach](#). *Preprint*, arXiv:2412.05655.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. [GPT-NER: Named entity recognition via large language models](#). *arXiv preprint arXiv:2304.10428*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Hugging-face’s transformers: State-of-the-art natural language processing](#). *Preprint*, arXiv:1910.03771.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Shan Zhang, Bin Cao, and Jing Fan. 2024. [KCL: Few-shot named entity recognition with knowledge graph and contrastive learning](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9681–9692.

## A Preliminary Self-Verification Experiments

We conducted preliminary experiments to assess the effect of the LLM self-verification step. Verification improved overall F1 for both Mistral-7B and LLaMA3-8B, mainly by increasing precision while slightly reducing recall. The effect was strongest for Mistral, whose weighted F1 increased from 0.375 to 0.409. For LLaMA3, the gain was smaller, with weighted F1 increasing from 0.473 to 0.481. However, verification produced larger gains for some lower-support labels. For instance, with LLaMA3, the F1-score for BATTERY\_CAPACITY increased from 0.291 to 0.552 and for BOOT\_SIZE from 0.112 to 0.213.

## B LLM Prompts

We use a per-entity extraction strategy following Wang et al. (2023). Each advertisement is processed once per entity type using the prompts below.

### Extraction Prompt

#### System:

You are a NER Model. Your task is to label {entity\_type} entities in the given sentence.

#### User:

Below are some examples.

{few\_shot\_examples}Input: {text}  
Output:

where each few-shot example follows the template:

Input: {input\_text}  
Output: {output\_text}

### Self-Verification Prompt

#### System:

You are a verification expert.  
Your task is to verify if an  
extracted entity is correct.

#### User:

Original text:

{text}

Entity type: {entity\_type}

Extracted entity: {entity}

Is this extraction correct and  
accurate? Consider:

1. Is the entity actually present  
in the text?
2. Is the entity type  
classification correct?
3. Are the boundaries accurate  
(no extra or missing characters)?

Answer only with a “yes” if all  
statements are true, otherwise  
say “no”. Do not say anything  
more.

Verification result:

vLLM (Kwon et al., 2023) on an NVIDIA RTX  
PRO 6000 GPU with 96GB VRAM. Larger mod-  
els were executed on an NVIDIA H200 SXM GPU  
with 141GB VRAM.

LLM inference included both the entity extrac-  
tion and entity verification stages. Processing the  
full evaluation set required approximately 1.5 hours  
per model. This corresponded to approximately  
 $109 \times 15 \times 2 = 3,270$  model queries, where 109  
is the number of test samples, 15 is the number of  
target entities, and 2 corresponds to the verification  
stage.

Closed-source models (e.g., GPT and Gemini)  
were accessed through their respective provider  
APIs. All LLMs were interfaced using an OpenAI-  
compatible API format. To ensure deterministic  
outputs, we set the temperature to 0.0 where sup-  
ported by the API and used greedy decoding for all  
generations.

## C Rules-based Precision and Recall

Table 9 show the per-label precision and recall  
scores for the best-performing rules-based meth-  
ods.

## D Encoder Training Set Size Analysis

Figure 2 shows the evaluation loss of the encoder  
models relative to the number of training samples.  
The plateauing of the loss curve suggests that the  
dataset size of 659 adverts was sufficient for the  
models to learn the primary patterns in the data in  
aggregate; as noted in Section 8, this does not hold  
uniformly across labels, and the sparsest categories  
remain under-supported.

## E Hardware and Runtime Details

All encoder-based transformer models were fine-  
tuned on an Apple M4 MacBook Pro equipped with  
36GB unified memory. Inference for the encoder-  
based models required between a few seconds and  
approximately one minute to process the complete  
test set.

For open-source LLMs, models with fewer  
than 10 billion parameters were executed using



Figure 2: Evaluation loss vs. number of training samples for encoder models.

| Label                        | Definition                                           | Example                                   | IAA  | <i>n</i> |
|------------------------------|------------------------------------------------------|-------------------------------------------|------|----------|
| <i>Vehicle identity</i>      |                                                      |                                           |      |          |
| MAKE                         | Vehicle manufacturer                                 | “CUPRA”                                   | .968 | 114      |
| MODEL                        | Specific model name                                  | “Focus”                                   | .962 | 178      |
| TRIM                         | Trim level / variant of a model                      | “220d Luxury”                             | .868 | 129      |
| YEAR                         | Registration year                                    | “2021”                                    | .967 | 61       |
| <i>Powertrain</i>            |                                                      |                                           |      |          |
| ENGINE_SPEC                  | Engine detail: volume, cylinders, or technology code | “TDCI”                                    | .912 | 104      |
| FUEL_TYPE                    | Fuel powering the vehicle                            | “petrol”                                  | .936 | 64       |
| TRANSMISSION                 | Gear system type                                     | “automatic”                               | .732 | 13       |
| <i>Body &amp; appearance</i> |                                                      |                                           |      |          |
| BODY_TYPE                    | Shape/style of the chassis                           | “Panel Van”                               | .867 | 20       |
| EXTERIOR_COLOUR              | Exterior paint colour, incl. descriptors             | “Black sapphire metallic”                 | .866 | 25       |
| INTERIOR_COLOUR              | Colour/material of interior features                 | “black of interior leather seats”         | .639 | 16       |
| NO_SEATS                     | Total seating capacity                               | “5 seats”                                 | .769 | 4        |
| BOOT_SIZE                    | Boot storage capacity (litres)                       | “6100 litres”                             | 1.00 | 1        |
| <i>Electric vehicle</i>      |                                                      |                                           |      |          |
| BATTERY_CAPACITY             | Battery capacity (kWh)                               | “68 kWh”                                  | .961 | 17       |
| BATTERY_RANGE                | Travel distance, incl. qualifiers                    | “max range of 280 miles when new”         | 1.00 | 10       |
| RECHARGE_TIME                | Recharge time, incl. charger/level context           | “empty to 80% in as little as 53 minutes” | .693 | 17       |

Table 1: The 15-label **AutoSpecNER** annotation schema, grouped by category with definitions and representative examples. IAA is the mean of the three pairwise inter-annotator F1 scores from the agreement study of Section 3.3.1; *n* is the mean number of instances of the label annotated per annotator in that study.

| Model                | Initial F1 | Verified F1 |
|----------------------|------------|-------------|
| Mistral-7B, macro    | 0.184      | 0.211       |
| Mistral-7B, weighted | 0.375      | 0.409       |
| LLaMA3-8B, macro     | 0.204      | 0.209       |
| LLaMA3-8B, weighted  | 0.473      | 0.481       |

Table 8: Effect of self-verification in preliminary LLM experiments.

Table 9: Precision and Recall Scores for Best Rules-Based Methods.

| Label           | Precision | Recall |
|-----------------|-----------|--------|
| battery         | 1.000     | 0.300  |
| model           | 0.880     | 0.775  |
| body_type       | 0.824     | 0.500  |
| make            | 0.666     | 0.949  |
| year            | 0.631     | 0.997  |
| trim            | 0.575     | 0.584  |
| fuel_type       | 0.479     | 0.717  |
| engine_spec     | 0.295     | 0.289  |
| transmission    | 0.197     | 0.655  |
| number_of_seats | 0.154     | 0.800  |
| exterior_colour | 0.015     | 0.134  |
| interior_colour | 0.004     | 0.064  |