AutoSpecNER: A Dataset and Benchmark for Named Entity Recognition of Vehicle Specifications

Anonymous submission

Abstract

Online text related to the automotive domain is ever-expanding through websites like car-selling platforms, technical forums and social media commentary; however there is no systematic way to track automobile mentions and their specifications. We introduce **AutoSpecNER**, a new expert-annotated dataset for fine-grained named entity recognition in the automotive domain. The dataset consists of 659 vehicle advertisements of a popular vehicle-selling website. It has over 10 thousand annotated entities across 15 distinct categories (e.g., `MODEL`, `ENGINE_SPEC`, `BATTERY_CAPACITY`). The annotation schema and data quality were validated through an inter-annotator agreement, achieving an average score of 91.5%. To benchmark the machine learning prediction in our dataset, we compared three distinct methodologies: a rules-based approach, fine-tuned transformer encoders (BERT, RoBERTa, DeBERTa, ModernBERT), and large language decoder models, open-sourced (Qwen, Llama, Mistral, Gemma) and closed-sourced (GPT, Gemini). Our results demonstrate that transformer-based encoders achieve the best performance, with DeBERTa reaching a micro F1-score of 90%, significantly outperforming the rules-based baseline (43%) and the top-performing large language model approach (77.8%). We commend this dataset for tracking mentions of car models and their specifications online and using it as a spurious automated content generation tool, contributing to the automotive NLP.

Keywords: Text Mining, Information Extraction, Named Entity Recognition, Automotive, Language Resources

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

Previous work by Ventirozos et al. (2024) demonstrated NER’s applicability to vehicle advertisements but focused on coarse-grained transactional categories (sales options, historic events, vehicle condition). We extend this work by developing a fine-grained annotation schema specifically for vehicle identifiable specifications, enabling applications such as automated specification tables, cross-validation between text and structured data, and hallucination detection in AI-generated content.

This paper presents two main contributions. First, we introduce Automotive Specification NER (**AutoSpecNER**), a new, publicly available dataset[1] 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. Unlike existing NER resources which focus on general domains like news (CoNLL-2003, Tjong Kim Sang and De Meulder (2003)) or noisy social media text (WNUT, Derczynski et al. (2017)), AutoSpecNER provides a specialised, high-quality resource for a specific commercial sector.

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

[1] Available at `[PLACEHOLDER]`

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 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 identifiables 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 labeled automotive data. While LLMs demonstrate competitive performance through prompting (Wang et al., 2023), recent studies (Naguib et al., 2024) show that smaller, specialized 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 collected 659 vehicle advertisements from one of the UK's biggest online vehicle-selling websites, which hosts everyday around half-million cars for selling. The dataset comprises two distinct sources:

User-generated advertisements (350): Written by individual sellers and dealerships, these contain natural language variations, informal descriptions, typos, and inconsistent formatting. Example: *"lovley ford focus 1.8 diesal, 2015 plate, full mot till next yr, grey metalic paint"*.

AI-generated advertisements (309): Created using Google Gemini[2] and Meta's LLaMA3[3] 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:

> "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 (ty-

[2] gemini-2.0-flash-001

[3] llama-3.1-8b-instruct

pos, 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, 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:

- **Annotate Faithfully to the Text, Not the Specification:** Labels must be applied to the text as it appears, even if it contradicts known vehicle specifications (i.e., a hallucination). For example, a trim level described in the text as "220d Luxury" was annotated as such, despite the official specification listing it only as "Luxury". This principle ensures the model learns to extract information present in the free text itself.

- **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.

3.3. Annotation Schema Overview

The final schema consists of 15 distinct entity labels. Below is a detailed reference for each label, including its definition and a representative example.

MAKE The manufacturer of the vehicle. *Example: "CUPRA"*

MODEL The specific model name of the vehicle. *Example: "Focus"*

TRIM The specific trim level or variant of a model. *Example: "220d Luxury"*

YEAR The registration year of the vehicle. *Example: "2021"*

ENGINE_SPEC Specific details of the engine, such as its volume, cylinder count, or technology code. *Example: "TDCI"*

FUEL_TYPE The type of fuel used to power the vehicle. *Example: "petrol"*

TRANSMISSION The gear system type. *Example: "automatic"*

BODY_TYPE The shape or style of the vehicle's chassis. *Example: "Panel Van"*

EXTERIOR_COLOUR The colour of the vehicle's exterior paint, including any descriptive terms. *Example: "Black sapphire metallic"*

INTERIOR_COLOUR The colour and material of interior features, such as seats or dashboards. *Example: "black leather seats"*

NUMBER_OF_SEATS The total seating capacity of the vehicle. *Example: "5 seats"*

BOOT_SIZE The storage capacity of the boot, typically in litres. *Example: "6100 litres"*

BATTERY_CAPACITY The electric capacity of an electric vehicle's battery, typically in kWh. *Example: "68 kWh"*

BATTERY_RANGE The distance an electric vehicle can travel, including any qualifying clauses. *Example: "maximum range of 280 miles when new"*

RECHARGE_TIME The time taken to recharge an electric vehicle's battery, including context about charger type and charge level. *Example: "empty to 80% in as little as 53 minutes"*

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 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.

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).

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 1.

We evaluate all models—encoders, LLMs, and rules-based approaches—using character-level IOB2 tagging to ensure fair comparison across different tokenisation schemes and the text generated by the decoder-type language models. We compute the metrics using the `seqeval` library (Nakayama, 2018).

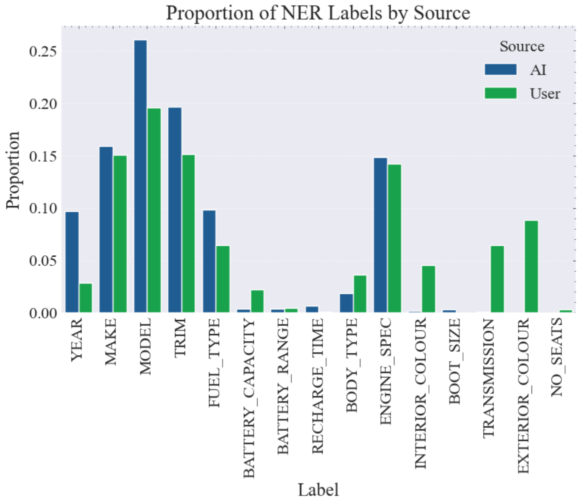

Figure 1: Proportion of labels from each source within the dataset. In dark blue are the the text advertisements generated by either Gemini or Llama and in green are the ones written by users.

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	260	78	10
INTERIOR_COLOUR	178	16	107
BODY_TYPE	116	32	33
TRANSMISSION	114	9	16
BATTERY_CAPACITY	86	44	32
BATTERY_RANGE	71	41	3
BOOT_SIZE	16	8	12
NO_SEATS	4	0	6
Overall	**7539**	**1518**	**2060**

Table 1: Distribution of entity counts across the final training, validation, and testing sets.

4.1. Models

We evaluated three classes of models on the AutoSpecNER 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 normalized 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 2, were designed to be robust enough to capture common formulations of these attributes.

4.1.2. Transformer Encoders

We fine-tuned three 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., 2021). 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 2×10^{-5}, batch size 8, weight decay 0.01, with 100 warmup steps and a maximum sequence length of 512 tokens.

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 Gemma et al., 2025). Additionally, we evaluated two closed-sourced LLMs: Gemini-2.5 (Gemini Team et al., 2025) and GPT-5 (OpenAI, 2025). 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 3.

Prompt Engineering We adopted the GPT-NER methodology (Wang et al., 2023), implementing a 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.

Complete prompt templates are provided in Appendix [PLACEHOLDER].

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

Table 2: Regular expression patterns used for specific entity extraction.

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 4 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[4] 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, the best-performing configuration of the rules-based approach achieved a micro F1-score of 0.43.

5.2. Encoder-based Models

We evaluated four transformer-based encoders: BERT, RoBERTa, ModernBERT and DeBERTa.

Table 3 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.

The DeBERTa model demonstrated the best overall performance, which is likely attributable to its disentangled attention mechanism better capturing token positions, a feature particularly relevant for the consistently formatted AI-generated adverts.

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

Table 4 shows per-label F1 scores for all the family models. The encoders perform best on numeric entities such as **YEAR** (98.2% F1) and **BATTERY_CAPACITY** (100% F1), standardized attributes like **FUEL_TYPE** (95.4% F1) and **TRANSMISSION** (93.8% F1), and vehicle identifiers including **MAKE** (93.4% F1) and **MODEL** (94.9% F1). In contrast, they struggle with more subjective attributes such as **INTERIOR_COLOUR** (57.5%–62.3% F1) and **EXTERIOR_COLOUR** (69.6%–72.5% F1), likely due to the wide lexical variation in colour descriptions (e.g., midnight blue", pearl white", "metallic silver").

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 [PLACEHOLDER]. The one exception is the `NO_SEATS` which was scarcely reported in the text descriptions.

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.

As expected, the self-verification step consistently increased precision while reducing recall. The performance increase from self-verification was most significant in lower-support labels with lower initial performance metrics. 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.

Gemini-2.5 scored the highest amongst the LLMs but still second to the encoders. The ana-

[4] Full Precision and Recall scores for all rules-based methods are available in Appendix [PLACEHOLDER].

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

Table 4: Per-label F1 scores comparing encoder, LLM, and rules-based performance. In bold are the top performants for each label. Some model names are shortened for illustration purposes but are the same version with the ones presented in Table 3.

lysis shows that its predictions across the test set reveals heterogeneous performance across entity types. LLMs demonstrate strong performance (F1 $>$ 75%) on entities including YEAR (94.8% F1), MAKE (79.1% F1), MODEL (87.8% F1). Moderate performance (F1 50–75%) is observed for ENGINE_SPEC (73.0% F1), EXTERIOR_COLOUR (69.4% F1), TRIM (71.5% F1), BATTERY_CAPACITY (71.4% F1), and BODY_TYPE (66.7% F1). Poor performance (F1 $<$ 50%) occurs on INTERIOR_COLOUR (27.0% F1), TRANSMISSION (32.0% F1), and notably RECHARGE_TIME (0% F1 despite 10 test instances).

LLMs exhibit three 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, 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.

5.3. Computational Resources

All encoder-based transformer models were fine-tuned on an Apple M4 MacBook Pro with 36GB unified memory. Inference for encoder models completed between seconds to minute for the entire test set. For LLM experiments, we distinguished between open-source and closed-source models. Smaller open-source LLMs (parameters $<$10B) performed inference using vLLM (Kwon et al., 2023) on an NVIDIA RTX PRO 6000 GPU with 96GB VRAM, while larger models utilized an NVIDIA H200 SXM GPU with 141GB VRAM memory. LLM inference, including the entity extraction and verification steps, required approximately 1.5 hours for each LLM to process all test samples (that is for 109 [n. samples] $\times$ 15 [number of entities] $\times$ 2 [verification test] number of queries). All closed-source LLMs (i.e. GPT, Gemini) were accessed through their respective provider APIs. All LLMs used the OpenAI-compatible API format. To ensure deterministic outputs, we set temperature to 0.0 where supported by the API and employed greedy decoding for all LLM generations.

6. Discussion

The current investigation has implications for deploying NER systems in production automotive applications. Despite their need for training, fine-tuned encoder models remain the best solution overall when the goal is extracting entity types and populating structured vehicle databases. The LLMs performance followed and last were the rule-based approaches. The LLMs with minimal prompting perform reasonably well, but the fine-tuned encoders, with DeBERTA leading overall, performed best when the supporting examples are enough.

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 challenge of extracting fine-grained vehicle specifications from the text descriptions of car advertisements. This task can have a high impact in the automotive world. Our dataset and annotation schema **AutoSpecNER** comprises 659 advertisements written by real users and AI-generated proprietary data acquired by the UK's biggest online vehicle selling website. The schema consists of 15 different vehicle specification labels ranging from `MAKE`, `MODEL`, to `RECHARGE_TIME`. We did an inter-annotator evaluation to validate the schema, which resulted in a commendable 91.5% F1 score. Next, we evaluated with various approaches: rule-based, fine-tuned encoder transformers and few-shot decoder transformers. The fine-tuned transformer encoders are exceptionally well-suited for this task, with DeBERTa achieving a top micro F1-score of 0.901. The decoder models followed with Gemini-2.5 being the first amongst them, and lastly the rule-based approaches. The LLMs are deemed particularly resourceful for low-support labels, which are scarcely reported in the text. Lastly, this dataset can be resourceful for market trend analysis and crucial for detecting 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 assessment of cross-lingual performance. Improving generalisability would require a more heterogeneous, multi-source, and multilingual dataset.

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.

10. Bibliographical 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 163–173.

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.

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, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, and Xi Chen et al. (1251 additional authors not shown). 2025. Gemini: A family of highly capable multimodal models.

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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Cauchetaux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, and Junteng Jia et al. (460 additional authors not shown). 2024. The llama 3 herd of models.

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, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Songhua Hu and Ruhu 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. IEEE.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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*, pages 94–100.

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.

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 Aurélie Névéol. 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. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

OpenAI. 2025. Introducing gpt-5. Product announcement, August 7.

Jiho Park, Jiseong Lee, Seonghyeon Kim, and Chanjun Park. 2023. ADMit: Improving NER in automotive domain with domain adversarial training and multi-task learning. *Expert Systems with Applications*, 225:120007.

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.

Team Gemma, 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, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak

Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, and Ivan Nardini (et al. 116 additional authros not shown). 2025. Gemma 3 technical report.

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.

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, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *arXiv 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.