International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Dependency-guided aspect extraction with coreference resolution and binary aspect classification

Reneilwe Kopane [1,*] and Mamello Monyane [2]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China.
[2] School of Computing, University of South Africa, South Africa.

International Journal of Science and Research Archive, 2026, 19(01), 402-414

Publication history: Received on 01 March 2026; revised on 06 April 2026; accepted on 09 April 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.1.0748

Abstract

Aspect-Based Sentiment Analysis (ABSA) aims to identify opinion targets (aspects) in text and determine the sentiment expressed toward each of them. Classical ABSA models based on recurrent networks and transformers achieve strong performance on benchmarks, but they often under-extract explicit aspects in syntactically complex sentences and do not generalize well across domains. Pure prompt-based use of Large Language Models (LLMs) offers strong semantic reasoning but tends to miss aspects, is sensitive to prompt design, and incurs non-trivial inference cost. This paper proposes CoreDep, a dependency-driven hybrid framework that integrates coreference resolution, dependency parsing, and a fine-tuned transformer classifier for robust aspect extraction. CoreDep first generates a high-recall candidate set using linguistic structure, then applies a binary classifier to recover precision. We evaluate aspect extraction under fuzzy span matching to account for boundary ambiguity, and demonstrate that the extracted aspects support effective downstream sentiment analysis using both LLMs and supervised classifiers. Experiments on the SemEval-2014 Laptop dataset show that CoreDep achieves an aspect-level precision of 0.8448, recall of 0.7508, and F1-score of 0.7950. We further construct and annotate a new laptop review dataset from Takealot.com containing 3,175 aspect instances, on which our domain-adapted model achieves an F1-score of 0.7766. Overall, the results indicate that combining linguistic structure with lightweight neural validation remains an effective and practical strategy for ABSA in the LLM era.

Keywords: Aspect-Based Sentiment Analysis; Dependency Parsing; Coreference Resolution; Aspect Extraction; Binary Classification; Large Language Models

1. Introduction

User-generated reviews on e-commerce platforms provide rich information about user experience, product strengths, and recurring complaints. Traditional sentiment analysis reduces each review to a single polarity label, which ignores the fact that users often comment on multiple aspects such as *battery life, screen, keyboard, or customer service* in the same sentence. Aspect-Based Sentiment Analysis (ABSA) seeks to address this limitation by identifying fine-grained opinion targets and predicting sentiment for each target [1-3].

Early ABSA work relied on lexicons and hand-crafted rules to extract opinion words and corresponding targets [4-6]. With the rise of deep learning, neural architectures such as ATAE-LSTM, deep memory networks, and interactive attention networks improved the modelling of aspect–context interactions [7-9]. Recent transformer-based models such as BERT-SPC and its variants further improved performance by leveraging pre-trained contextual representations and attention mechanisms [10–13]. Although a large body of work has studied tasks, models, and challenges for ABSA, several practical issues remain.

* Corresponding author: Reneilwe Kopane

First, both classical and transformer-based ABSA models often under-extract aspects, especially in sentences with multiple targets or complex syntax. For example, in the laptop review sentence

"The battery life is amazing, but the screen is dull and the fan is noisy,"

a model may correctly detect *battery life* but miss *screen* or *fan*, leading to low recall. Second, aspect extraction often relies on exact-span supervision; small boundary differences such as "battery" versus "battery life" are treated as errors, which may not match downstream application needs. Third, models trained on standard benchmarks such as SemEval-2014 [14] do not automatically generalize to domain-specific platforms such as Takealot.com, a major South African e-commerce platform, with different writing styles and local expressions.

Meanwhile, LLMs such as GPT-3, GPT-4, and open-source LLaMA-style models [15-17] show strong zero-shot and few-shot abilities on many NLP tasks, including sentiment analysis [18-20]. Several recent works explicitly explore LLMs for ABSA [21–28]. These studies report that LLMs can identify aspects and sentiments without task-specific training, but they also highlight limitations: LLMs tend to under-predict the number of aspects, are sensitive to prompt design, and can hallucinate aspects that do not appear in the text.

At the same time, linguistic tools such as dependency parsers and coreference resolvers encode structural information that is highly relevant for ABSA [29–31]. Dependency relations like adjectival modifiers and subject–object links point to likely aspect–opinion pairs, while coreference resolution helps connect pronouns such as "it" or "this laptop" back to the correct entity [32–35]. However, purely rule-based use of these tools often yields noisy candidates and brittle heuristics [31].

Despite strong progress in ABSA, there remains a gap between high-capacity end-to-end neural models and the practical requirements of robust aspect extraction in real-world reviews. Existing transformer-based ABSA systems often suffer from low aspect recall in syntactically complex sentences, while pure LLM-based prompting tends to under-extract aspects and lacks

controllability. At the same time, linguistic signals such as dependency relations and coreference structure are known to be highly informative for aspect identification, but are typically used either heuristically or in isolation.

We hypothesize that explicitly leveraging dependency parsing and coreference resolution to generate a high-recall candidate set, followed by binary classifier-based candidate validation to recover precision, can yield more robust aspect extraction than either purely neural or purely rule-based approaches. Furthermore, we hypothesize that evaluating aspect extraction under fuzzy span matching better reflects the inherent boundary ambiguity of aspect terms and provides a more realistic assessment of downstream utility.

To test these hypotheses, we design CoreDep as a modular pipeline that separates candidate generation from candidate validation, and we evaluate it across two domains (SemEval-2014 and Takealot) using token-level Jaccard matching. This design allows us to directly analyse the contribution of linguistic structure, learned filtering, and domain adaptation within a unified framework.

Motivated by the identified gaps, we seek a method that:

- Exploits dependency structure and coreference information to generate high-recall aspect candidates;
- Uses a learned binary classifier to determine which candidates function as true aspects in context, rather than relying solely on hand-written rules;
- Can be adapted to a new domain with limited annotated data; and
- Remains compatible with LLM-based sentiment reasoning, so that aspect extraction and sentiment analysis can be combined in a larger pipeline.

To address the identified problem, we propose CoreDep, a dependency-driven hybrid framework for robust aspect extraction. Conceptually, CoreDep separates the problem into three layers:

- Linguistic layer: We normalize text, apply coreference resolution, and parse dependency trees. From each sentence, we derive a high-recall set of candidate aspect spans based on syntactic patterns such as noun chunks and opinion-bearing contexts.

- Classification layer: Candidate spans produced by the linguistic layer are evaluated in sentence context by a binary classifier, which predicts whether each span is a valid product aspect or a non-aspect candidate.
- LLM reasoning layer: The resulting aspect inventory can be fed into an LLM such as GPT-4 to perform sentiment classification or to generate structured summaries; this paper focuses on the extraction side and designs the pipeline to be LLM-compatible.

We formulate aspect validation as a binary classification problem in which each candidate is assigned one of two labels: aspect or non-aspect. Our experiments indicate that coreference resolution, dependency-driven candidate generation, and domain adaptation together yield a practical and effective aspect extraction system.

2. Materials and methods

Figure 1 presents the overall framework of the proposed method. The pipeline begins with review data from the SemEval-2014 and Takealot datasets, followed by coreference resolution and dependency parsing to generate linguistically grounded candidate aspects. These candidates are then passed to a binary classification-based validation stage, which separates valid aspect terms from non-aspect candidates. This design combines explicit linguistic structure with learned filtering, allowing the system to maintain high-recall during candidate generation while improving precision during refinement.

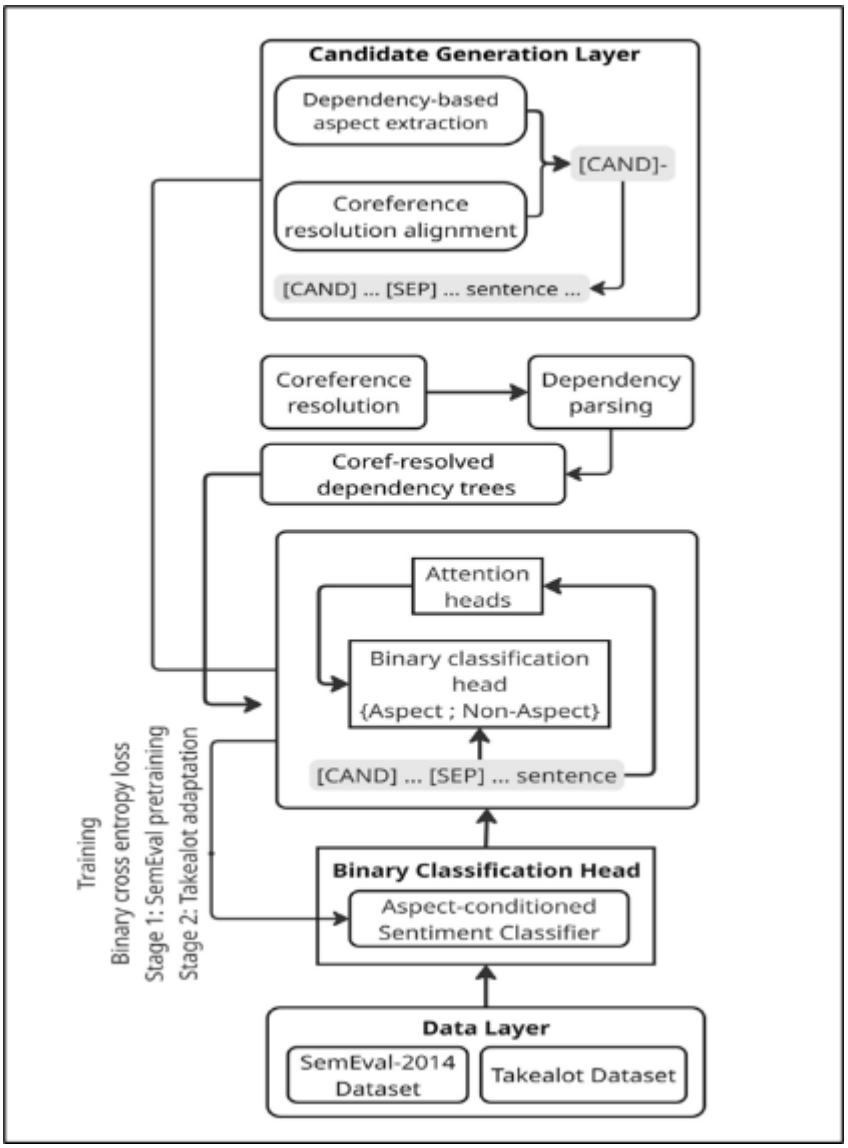

Figure 1 Overall framework of the proposed pipeline

This section describes the CoreDep framework in detail as presented in Figure 1. We first define notation, then explain each component: coreference resolution, dependency-based candidate generation, binary aspect classification, and fuzzy evaluation. Let s denote an input review sentence. The goal of aspect extraction is to recover the set

$$A(s) = a_1, a_2, \dots, a_{N_s}, \tag{1}$$

Where each a_i is contiguous span in s corresponding to an aspect term (e.g., "battery life" or "customer service"). In supervised settings, gold aspect spans are annotated in the training data [14]. Our framework decomposes this into two sub-problems:

- Candidate generation: Given s construct a set of candidate spans $C(s)$ such that $A(s) \subseteq C(s)$ as often as possible.
- Candidate validation: For each $\in C(s)$, decide whether c is a true aspect or not.

We then evaluate the predicted aspect set $A(s)$ against gold spans using fuzzy matching.

2.1. Coreference-aware preprocessing

Customer reviews frequently contain pronouns and underspecified references such as "it", "this laptop", or "this one", which can make downstream reasoning ambiguous. To reduce this ambiguity, we apply a neural coreference resolver to each sentence or short review segment. Let s denote the original sentence and $\hat{s}$ denote the coreference-resolved sentence, in which pronouns are replaced by their antecedents when appropriate

$$\hat{s} = \text{CorefResolve}(s). \tag{2}$$

We use a FastCoref-style model [34,35] that predicts coreference clusters and provides resolved text. This step ensures that candidate generation and classification operate on explicit entity mentions, which is particularly useful when the aspect term itself appears only once but is referenced multiple times via pronouns.

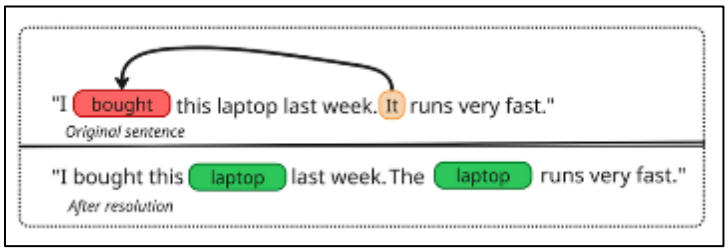

Figure 2 Example of coreference-aware normalization

2.2. Dependency-guided candidate generation

Given the resolved sentence $\hat{s}$ we apply a dependency parser to obtain a syntactic representation. We then generate a high recall candidate set $C(\hat{s})$ using noun centred spans and lightweight dependency informed expansions, following common practices in dependency-based opinion mining and ABSA [30-32]. In particular, we include:

- Noun chunks: base noun phrases headed by common nouns or proper nouns (e.g., *battery life*, *touch screen*);
- Compound expansions: multi-word compounds attached to a noun head (e.g., *power button*, *Windows license*);
- Predicate-linked nouns: noun targets connected to opinion predicates via subject/object relations (e.g., *love the keyboard*, *hate this trackpad*);
- Copular subjects: nouns serving as subjects in copular constructions (e.g., *the screen is bright*, *the fan is loud*).

Let P denote this set of extraction rules/patterns. For each resolved sentence $\hat{s}$, the the candidate set is:

$$C(\hat{s}) = \bigcup_{p \in P} p(\hat{s}) \tag{3}$$

where $p(\hat{s})$ returns the spans matched by pattern p. By design, $C(\hat{s})$ is intentionally over-complete: it may contain generic nouns, partial spans, and non-aspect mentions. Therefore, we learn a neural validator to recover precision by filtering spurious candidates in context.

Candidate aspect extraction is formulated as a rule-matching process over dependency edges. Each rule specifies: (i) a target part-of-speech tag for the aspect token, (ii) a related token type, and (iii) a dependency relation linking them. Formally, a rule R_k is defined as:

$$R_k = (pos_a, pos_m, dep_r) \tag{4}$$

where pos_a is the POS tag of the candidate aspect, pos_m is the POS tag of a related modifier or action token, and dep_r is the dependency relation connecting them. A candidate aspect a is extracted if there exists a token pair (w_i, w_j) such that:

$$POS\ (w_i) = pos_a \wedge POS(w_j) = pos_m \wedge dep((w_i), w_j) = dep_r \tag{5}$$

This formulation enables deterministic identification of aspect–modifier or aspect–action pairs from dependency trees. Based on linguistic patterns commonly observed in product reviews, a set of high-yield dependency rules is designed to guide extraction. Table 3.2 summarizes the primary rules used in this paper.

Each extracted noun satisfying one or more rules is added to the candidate aspect list. When multiple nouns are connected via compound or conjunct relations, they are merged to form multi-word aspects such as *battery life* or *screen quality.* Consider the review sentence:

> "The battery life of this laptop lasts 8 hours."

The dependency parser produces relations:

- Compound(battery, life)
- Nmod:poss(life, laptop)
- Nsubj(lasts, life)
- Nummod(hours, 8)

Applying Rules R1–R8:

- Compound merges *battery* + *life* → battery life
- Nmod:poss links *battery life* to *laptop*
- Nummod attaches duration *8 hours*

Thus, the extracted candidate aspect is:

Aspect = battery life, *Context* = laptop, *Attribute* = 8 hours

Table 1 Core dependency rules

Rule	Aspect POS	Related POS	Dependency	Example
R1	NN	JJ	amod	battery → durable
R2	NN	VB	nsubj	screen → displays
R3	NN	VB	auxpass	battery → replaced
R4	NN	NN	nmod:poss	battery → laptop's
R5	NN	IN	prep+pobj	battery → in laptop
R6	NN	JJR	amod	battery → more durable
R7	NN	NUM	nummod	battery → 8 hours
R8	NN	NN	conj	keyboard, touchpad

2.3. Binary aspect classification

We formulate candidate validation as binary classification. For each candidate span $c \in C(\hat{s})$, we construct an input sequence

$$x = [CAND]c[SEP]\hat{s}, \tag{6}$$

and assign a label $y \in \{0.1\}$, where $y = 1$ if c matches a gold aspect span in $\hat{s}$ under fuzzy matching (during training), and $y = 0$ otherwise. A transformer encoder f_θ maps x to a vector representation $h(x) \in R^d$. A linear layer followed by a sigmoid produces the probability that c is a true aspect:

$$p_\theta(y = 1|x) = \sigma(w^\top h(x) + b), \tag{7}$$

where $w \in R^d, b \in R$, and $\sigma(\cdot)$ is the logistic function. We train f_θ by minimizing the binary cross-entropy over annotated candidates:

$$L(\theta) = \sum_i [yi\ log\ p_\theta(y_i = 1 \mid x_i) + (1 - y_i)\ log(1 - p_\theta(y_i = 1 \mid x_i))] \tag{8}$$

At inference time, a candidate c is accepted as an aspect if $p_\theta(y = 1 \mid x) \geq \tau$, where τ is is a validation threshold. The final predicted aspect set for sentence s is

$$\hat{A}(s) = c \in C(\hat{s}) | p_\theta(y = 1|x) \geq \tau. \tag{9}$$

Figure 3 illustrates the internal architecture of the aspect validation model. The input sequence, constructed as $[CAND]$ candidate $[SEP]$ sentence, is encoded by a DeBERTa-v3 transformer. The contextualized representations of the candidate span and $[CLS]$ token are pooled and passed through a small feed-forward network with GELU activation and dropout to predict whether the candidate is a true aspect. This architecture is trained first on SemEval (Stage 1) and then adapted to Takealot (Stage 2) through continued fine-tuning.

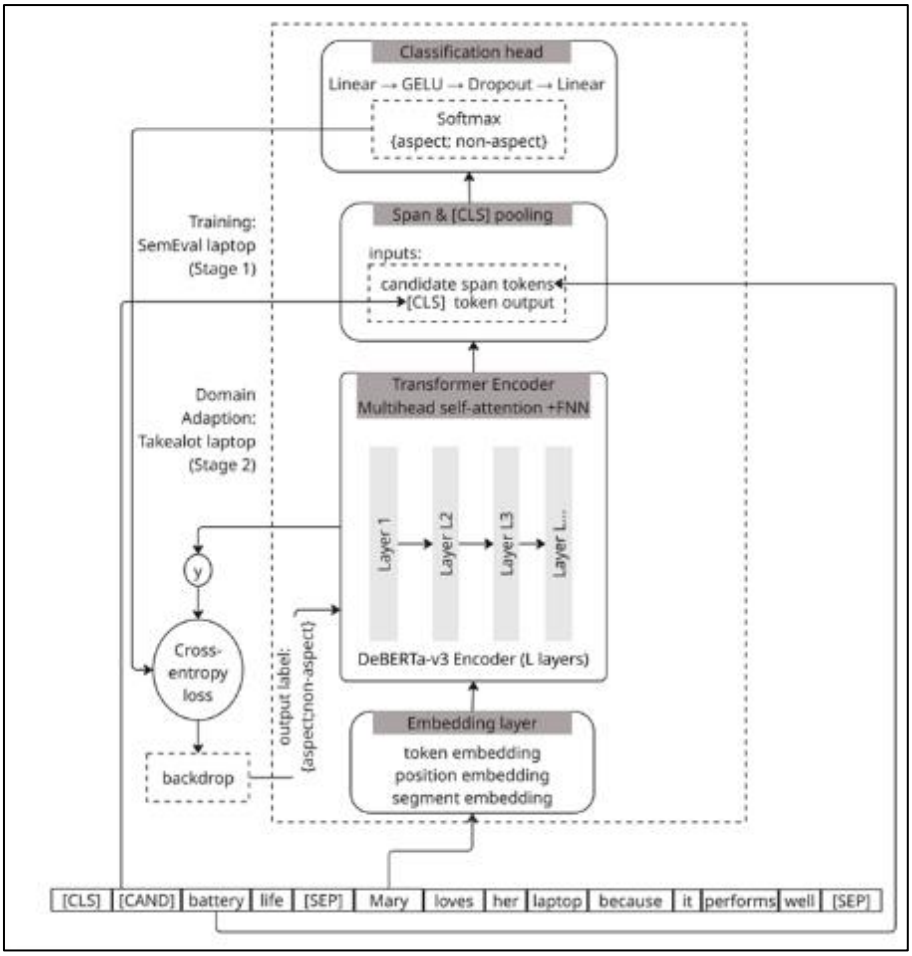

Figure 3 Architecture of the proposed Aspect Candidate Classifier

2.4. Fuzzy matching and evaluation

Exact character-level matching penalizes small boundary mismatches that are often harmless for downstream analysis. We therefore adopt a token-level Jaccard similarity to judge whether a predicted span â matches a gold span a:

$$J(a, \hat{a}) = \frac{|T(a) \cap T(\hat{a})|}{T(a) \cup T(\hat{a})}, \tag{10}$$

where $T(\cdot)$ denotes the set of lowercase tokens. A pair $(a, \hat{a})$ is considered a match if $J(a, \hat{a}) \geq \gamma$, with $\gamma = 0.5$ Let TP, FP, and FN denote the number of true positives (matched predicted aspects), false positives (predicted aspects with no matching gold span), and false negatives (gold aspects with no matching prediction). The aspect-level precision, recall and F1 are computed as:

$$Precision = \frac{TP}{TP + FP}, \tag{11}$$

$$Recall = \frac{TP}{TP + FN}, \tag{12}$$

-

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{13}$$

These metrics are reported at the dataset level by aggregating counts over all sentences.

3. Results

3.1. Dataset characteristics

The experiments were conducted on two aspect-level datasets: the SemEval-2014 Laptop benchmark and a Takealot laptop review dataset. These datasets differ not only in size but also in linguistic properties. SemEval represents a controlled benchmark with relatively clean annotations, whereas the Takealot dataset contains noisier, real-world e-commerce reviews characterized by informal language, spelling variation, and domain-specific expressions. We use the SemEval-2014 Task 4 laptop review dataset [15] as a benchmark. Each sentence is annotated with one or more aspect terms and sentiment polarities. We collected laptop reviews from Takealot.com by scraping product review pages. All reviews were filtered to retain English-language content and to remove duplicates, empty entries, and boilerplate text. Reviews were segmented into sentences, and annotation was performed at the aspect-instance level rather than the sentence level. From the filtered reviews, we annotated a total of 3,175 aspect instances. Each instance corresponds to a tuple (s, a, y), where s is a review sentence, a is an explicit aspect term appearing in s, and $y \in \{positive, neutral, negative\}$ is the sentiment expressed toward that aspect. A single sentence may contain multiple annotated aspects, which results in multiple aspect instances derived from the same sentence. Aspect terms are defined as contiguous text spans that explicitly denote an opinion target related to the laptop or closely associated service attributes (e.g., battery life, screen, fan noise, delivery service). Annotators were instructed to select the minimal span that preserves the semantic identity of the target. Generic nouns (e.g., price, quality) were annotated only when the surrounding context clearly indicated a specific, evaluable attribute.

Each aspect instance was labelled as $positive, negative, or\ neutral$. The neutral label was used for factual statements, mixed sentiment, or cases without a clear evaluative judgment. Annotation followed SemEval-style ABSA guidelines [16], and inconsistencies in aspect boundaries or sentiment labels were resolved through manual adjudication by the first author to ensure internal consistency.

For this paper we focus on aspect extraction and use the provided train/test split. Following common practice [11,13], we pre-process the data by lowercasing and removing obvious noise but retain the original aspect spans for evaluation. Table 2 summarizes the dataset statistics after preprocessing and splitting.

Table 2 Dataset statistics

Dataset	Total	Train	Dev	Test	Positive	Negative	Neutral
SemEval-2014 Laptop	2313	1850	231	232	987	866	460
Takealot	3175	2540	317	318	1371	1081	723

3.2. Aspect candidate classification and extraction

3.2.1. Candidate classification

The performance of the binary candidate classifier is summarized in Table 3. The model achieves consistently high accuracy and macro-F1 across both datasets, indicating stable and well-balanced classification between aspect and non-aspect candidates. The high aspect-class F1 further confirms that the classifier effectively identifies valid aspect spans from the noisy candidate pool generated during the dependency-based extraction stage. Overall, these results demonstrate that the classifier serves as a reliable validation layer, improving precision without sacrificing the diversity of candidates introduced by the high-recall generation process.

Table 3 Candidate classification and aspect extraction performance

Dataset	Candidate classifier Accuracy	Candidate classifier Macro-F1	Aspect class F1	Extraction Precision	Extraction Recall	Extraction F1
SemEval-2014	0.92	0.92	0.91	0.8448	0.7508	0.7950
Takealot	0.93	0.93	0.92	0.8146	0.7421	0.7766

3.2.2. Aspect extraction

The final aspect extraction performance of the proposed CoreDep pipeline is also presented in Table 3, evaluated using token-level Jaccard similarity for fuzzy span matching. This evaluation strategy reduces sensitivity to minor boundary mismatches and better reflects the practical usefulness of extracted aspect spans. As shown in Table 3, the model maintains a strong balance between precision and recall across both datasets, indicating that the combination of dependency-guided candidate generation and classifier-based validation effectively controls noise while preserving relevant aspect terms.

Table 4 compares CoreDep with representative baseline models, including transformer-based approaches such as BERT-SPC [11,12], dependency-based models such as ASGCN, toolkit-based extraction using PyABSA [37], and prompt-based LLM methods [22]. CoreDep achieves competitive F1 performance on both SemEval-2014 and Takealot, outperforming traditional neural models and prompt-based approaches. These results suggest that incorporating explicit linguistic structure with learned validation offers a robust advantage over purely end-to-end or purely rule-based extraction strategies.

Table 4 Baseline comparison and component contribution analysis

- Part A: Comparison with baselines

Model	**SemEval F1**	**Takealot F1**
BERT-SPC[11,12]	0.77	-
ASGCN	0.79	-
PyABSA (ATEPC)	0.78	-
GPT-based zero-shot prompting	0.76	-
CoreDep (ours)	**0.7950**	**0.7766**

- Part B: Component contribution analysis

Configuration	**SemEval F1**	**Takealot F1**
Full CoreDep	0.7950	0.7766
Without coreference resolution	0.782	0.760
Without dependency-based candidate generation	0.710	0.690
Without Takealot domain adaptation	0.7950	0.730
Rule-based filtering instead of classifier	0.750	0.720

3.3. Component contribution analysis

To understand the contribution of each component, we evaluate several variants of the pipeline. Table 4 Part B reports fuzzy aspect F1 on SemEval and Takealot when specific components are removed or modified. Removing coreference resolution causes a small but consistent drop, especially on Takealot, where pronouns and informal references to the laptop are frequent. Removing dependency-based candidate generation and relying solely on simple noun chunk heuristics leads to a significant decrease in F1, confirming that syntactic patterns are crucial for recall. Using only the SemEval-trained classifier on Takealot without domain adaptation yields noticeably lower F1, which highlights the importance of adapting to platform-specific language. Finally, replacing the learned classifier with hand-crafted rules also degrades performance, showing that the classifier effectively captures information beyond simple frequency or part-of-speech cues.

3.4. LLM aspect-conditioned sentiment classification

To evaluate whether the extracted aspects produced by CoreDep are suitable for downstream sentiment reasoning, we conduct an aspect-conditioned sentiment classification experiment using a LLM. Instead of relying on gold aspect spans,

sentiment is predicted based on the aspects extracted by our pipeline, reflecting a realistic deployment scenario. We construct a matched evaluation set by aligning predicted aspect spans with gold annotations using token-level Jaccard similarity with a threshold of γ_{align} =0.5. This results in a dataset of 1684 matched aspect instances, each represented as a tuple (s, a, y), where s is the review sentence, a is the extracted aspect, and $y \in \{positive, neutral, negative\}$ is the gold sentiment label. Given each sentence–aspect pair, the LLM is prompted to output exactly one sentiment label conditioned on the provided aspect. Table 6 reports the performance of our CoreDep pipeline combined with zero-shot LLM sentiment classification, compared with supervised transformer baselines. The LLM achieves an overall accuracy of 0.819 and a macro-F1 score of 0.780. Performance is strong for positive and negative sentiments, with F1 scores of 0.878 and 0.872 respectively. Neutral sentiment remains more challenging, achieving an F1 score of 0.589, which is consistent with prior observations in ABSA literature that neutral opinions are inherently harder to identify. These results demonstrate that the dependency-driven aspect extraction stage produces aspect representations that are sufficiently precise for effective LLM-based sentiment inference. Importantly, this decoupled design allows sentiment analysis to benefit from LLM reasoning without requiring end-to-end fine-tuning of large models, keeping the overall pipeline efficient and modular.

Among supervised baselines, DeBERTa-v3-base performs best. Compared with these supervised baselines, the zero-shot LLM achieves higher overall performance on the matched set, suggesting that LLM-based sentiment reasoning provides strong robustness for aspect-conditioned sentiment classification even without task-specific fine-tuning.

Table 5 Aspect-conditioned sentiment evaluation on extracted aspects

Method	Accuracy	Macro-F1	Negative F1	Neutral F1	Positive F1
CoreDep + LLM (zero-shot)	0.8189	0.7795	0.8719	0.5888	0.8778
BERT-base	0.7633	0.7206	—	—	—
RoBERTa-base	0.7396	0.7180	—	—	—
DeBERTa-v3-base	0.7929	0.7680	—	—	—

Figure 4 shows the confusion matrix of the LLM predictions. Rows correspond to gold sentiment labels and columns correspond to predicted labels.

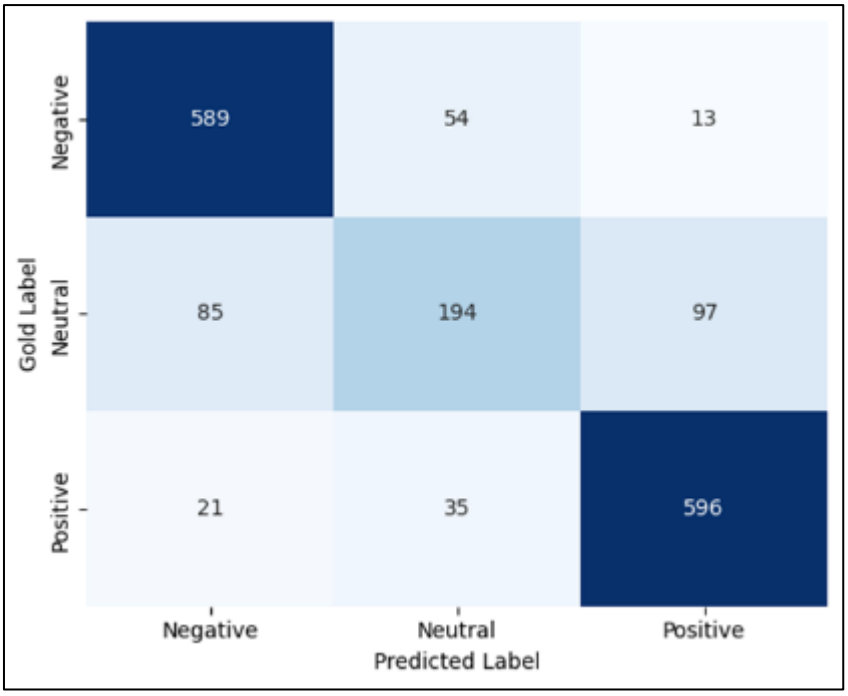

Figure 4 Confusion matrix for LLM-based aspect-conditioned sentiment

4. Discussion

We observe that CoreDep can extract nuanced laptop aspects such as Wi-Fi connectivity, pre-installed software, fan noise, and delivery speed on Takealot. For example, in the review

"The laptop is very portable, but the Wi-Fi keeps disconnecting and the battery dies quickly," the pipeline generates candidates such as *laptop*, *WiFi*, and *battery,* and the classifier correctly keeps the three device-related aspects. The fuzzy evaluation counts these as correct even if the predicted span is "battery" and the gold span is "battery life". The main remaining errors fall into three categories:

Overly generic nouns such as "price" or "quality" that sometimes appear in contexts where they are not true aspects of the laptop under the annotation guideline.

Highly implicit aspects where no clear noun phrase is present, and the sentiment is expressed indirectly (e.g., "It crashes all the time" without an explicit mention of stability).

Boundary ambiguity for multi-word aspects, especially when adjectives and compounds are stacked (e.g., "online customer service team").

Overall, the dependency-guided candidate set plus a strong classifier produces stable and interpretable performance. Compared with pure LLM prompting, the hybrid approach gives more control over what is considered an aspect and makes it easier to adapt to new domains or annotation schemes.

5. Conclusion

This paper presented CoreDep, a dependency-guided hybrid framework for aspect extraction in Aspect-Based Sentiment Analysis. The framework combines coreference resolution, dependency-based candidate generation, and a fine-tuned transformer classifier to identify explicit aspect spans in context. We evaluated CoreDep on the SemEval-2014 Laptop dataset and on a newly constructed and manually annotated Takealot laptop review dataset. Across both domains, CoreDep achieves strong aspect-level performance under fuzzy span matching, outperforming multiple transformer-based baselines and demonstrating the importance of linguistic structure and domain adaptation for robust aspect extraction. We further showed that the extracted aspects are suitable for downstream aspect-conditioned sentiment analysis. Using matched extracted aspects rather than gold spans, a zero-shot LLM achieves competitive performance compared with supervised non-LLM baselines, while neutral sentiment remains the most challenging class. These results highlight the value of decoupling aspect extraction from sentiment reasoning and positioning CoreDep as an LLM-compatible front-end that improves recall, controllability, and robustness in practical ABSA pipelines.

Future work will explore tighter integration between Core-Dep and LLMs, including using LLMs for candidate generation, re-ranking, or calibration. We also plan to extend the framework to multi-sentence and document-level settings using explicit coreference chains, and to investigate parameter efficient domain adaptation techniques such as lightweight adapters and LoRA. Finally, improving downstream sentiment labeling, particularly for neutral and mixed cases through calibration and confidence-aware model selection remains an important direction.

Compliance with ethical standards

Acknowledgments

The authors would like to acknowledge the support of the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. No external funding was reported for this study.

Disclosure of conflict of interest

The authors declare that there is no conflict of interest.

References

[1] Pang B, Lee L. Opinion mining and sentiment analysis. Found Trends Inf Retr. 2008;2(1-2):1-135.

[2] Liu B. Sentiment analysis and opinion mining. San Rafael (CA): Morgan & Claypool; 2012.

[3] Hu M, Liu B. Mining and summarizing customer reviews. In: Proc ACM SIGKDD Int Conf Knowl Discov Data Min; 2004. p. 168-177.

[4] Nazir A, Rao Y, Wu L, Sun L. Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. IEEE Trans Affect Comput. 2020;13(2):845-863.

[5] Zhang L, Wang S, Liu B. Deep learning for aspect-based sentiment analysis: A survey. IEEE Trans Knowl Data Eng. 2022;34(7):3374-3399.

[6] Hua YC, Denny P, Wicker J, Taskova K. A systematic review of aspect-based sentiment analysis: Domains, methods, and trends. Artif Intell Rev. 2024;57(11):296.

[7] Wang Y, Huang M, Zhu X, Zhao L. Attention-based LSTM for aspect-level sentiment classification. In: Proc 2016 Conf Empirical Methods in Natural Language Processing; 2016. p. 606-615.

[8] Tang D, Qin B, Liu T. Aspect level sentiment classification with deep memory network. In: Proc 2016 Conf Empirical Methods in Natural Language Processing; 2016. p. 214-224.

[9] Ma D, Li S, Zhang X, Wang H. Interactive attention networks for aspect-level sentiment classification. In: Proc 26th Int Joint Conf Artif Intell; 2017. p. 4068-4074.

[10] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv [Preprint]. 2019.

[11] Hoang M, Bihorac OA, Rouces J. Aspect-based sentiment analysis using BERT. In: Proc 22nd Nordic Conf Comput Linguistics; 2019. p. 187-196.

[12] Sun C, Huang L, Qiu X. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In: Proc 2019 Conf North Am Chapter Assoc Comput Linguistics Hum Lang Technol; 2019. p. 380-385.

[13] Huang X, Li J, Wu J, Chang J, Liu D, Zhu K. Flexibly utilizing syntactic knowledge in aspect-based sentiment analysis. Inf Process Manag. 2024;61(3):103630.

[14] Pontiki M, Galanis D, Pavlopoulos J, Papageorgiou H, Androutsopoulos I, Manandhar S. SemEval-2014 Task 4: Aspect based sentiment analysis. In: Proc 8th Int Workshop on Semantic Evaluation (SemEval 2014); 2014. p. 27-35.

[15] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. arXiv [Preprint]. 2020.

[16] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI technical report; 2019.

[17] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: Open and efficient foundation language models. arXiv [Preprint]. 2023.

[18] Wang Z, Xie Q, Feng Y, Chen Z, Yu Z, Wu L. Is ChatGPT a good sentiment analyzer? A preliminary study. arXiv [Preprint]. 2023.

[19] Kheiri K, Karimi H. SentimentGPT: Exploiting GPT for advanced sentiment analysis and its departure from current machine learning. arXiv [Preprint]. 2023.

[20] Mughal N, Mujtaba G, Shaikh S, Kumar A, Daudpota SM. Comparative analysis of deep neural networks and large language models for aspect-based sentiment analysis. IEEE Access. 2024;12:60943-60959.

[21] Zhou C, Song D, Tian Y, Wu Z, Wang H, Zhang X, et al. A comprehensive evaluation of large language models on aspect-based sentiment analysis. arXiv [Preprint]. 2024.

[22] Li X, Sun A. Prompting large language models for aspect-based sentiment analysis. arXiv [Preprint]. 2023.

[23] Wen Z, Zhu X, Wang P, et al. A hybrid approach to aspect-based sentiment analysis using transfer learning and large language models. Electronics. 2024;13(18):3724.

[24] Ding Y, Zhang H, Liu C, et al. Boosting large language models with continual learning for aspect-based sentiment analysis. arXiv [Preprint]. 2024.

[25] Neveditsin V, Ruder S, Rei M. From annotation to adaptation: Metrics, synthetic data, and aspect extraction for aspect-based sentiment analysis with large language models. In: Proc NAACL; 2025.

[26] Silva A, Pinho C, Santos R. Large language models for aspect-based sentiment analysis in tourism reviews. Expert Syst Appl. 2025;261:125514.

[27] Simmering PF, Huoviala P. Large language models for aspect-based sentiment analysis. arXiv [Preprint]. 2023.

[28] Zhang Y, Wen S, Zhu Y, et al. Combining large language models with interpretable models for explainable aspect-based sentiment analysis in the medical domain. J King Saud Univ Comput Inf Sci. 2025;37:175.

[29] Poria S, Cambria E, Winterstein G, Huang GB. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. In: Proc COLING; 2014. p. 268-278.

[30] Sun A, Li X, He Y. Exploiting dependency structures for aspect-level sentiment analysis. Inf Sci. 2021;579:138-151.

[31] Mishra P, Panda SK. Dependency structure-based rules using root node technique for explicit aspect extraction from online reviews. IEEE Access. 2023;11:65117-65137.

[32] Gupta P, Poria S, Gelbukh A, Cambria E. A neural architecture for opinion-aware coreference resolution. In: Proc COLING; 2018. p. 938-949.

[33] Lee K, He L, Lewis M, Zettlemoyer L. End-to-end neural coreference resolution. In: Proc 2017 Conf Empirical Methods in Natural Language Processing; 2017. p. 188-197.

[34] Jiang Z, Li L, Zheng Y, Neubig G. FastCoref: Fast and accurate coreference resolution. In: Proc NAACL-HLT; 2023.

[35] Otmazgin S, Cattan A, Goldberg Y. F-coref: Fast, accurate and easy to use coreference resolution. In: Proc AACL-IJCNLP 2022 System Demonstrations; 2022. p. 147-157.

[36] Gururangan S, Marasović A, Swayamdipta S, Lo K, Beltagy I, Downey D, et al. Don't stop pretraining: Adapt language models to domains and tasks. In: Proc ACL; 2020. p. 8342-8360.

[37] Huang L, Sun C, Qiu X. A hybrid neural architecture for aspect-based sentiment analysis. Neurocomputing. 2021;427:215-225.