Beyond Benchmarks: Interpretable Aspect-Based Sentiment Analysis for Actionable Insights from Customer Reviews

Anonymous for Review

***Abstract*—The pursuit of advanced performance in Aspect-Based Sentiment Analysis has led to a dependence on complex, large-scale generative models. This trend creates a significant disconnect between academic benchmarks and the practical needs of enterprise applications, which require transparent, auditable, and actionable insights. This paper introduces Syntactic Pattern Aspect-Based Sentiment Analysis (SP-ABSA), a lexicon-driven framework designed to bridge this interpretability gap. Our primary contribution is a novel unsupervised two-stage methodology. In the first stage, an offline process automatically constructs a domain-specific aspect lexicon from the raw corpus using noun chunk extraction and a hybrid semantic-morphological clustering algorithm. This lexicon then guides the second, online stage, which utilises explicit syntactic pattern checkers on dependency-parsed text to extract sentiment triplets in the format [Aspect, Attribute, Sentiment]. When evaluated against the SemEval-2014 benchmark, SP-ABSA achieved an F1-score of 0.215, showing a significant improvement over a traditional dependency-rule baseline score of 0.155. More importantly, our qualitative analysis highlights a systemic divergence between the framework's exhaustive, grammatically grounded outputs and the subjective, salience-focused nature of benchmark annotations. By systematically identifying instances of 'correct but unannotated' insights, we argue that standard metrics are insufficient for evaluating discovery-oriented systems.**

***Index Terms*—Aspect-Based Sentiment Analysis, Interpretability, Explainable AI, Rule-Based Systems, Natural Language Processing, Actionable Intelligence**

I. Introduction

Aspect-Based Sentiment Analysis (ABSA) offers a nuanced approach to understanding user-generated content, shifting the focus from general sentiment to specific opinions regarding individual features of a product or service [1]. This attention to detail is essential, as traditional document-level analysis frequently fails to capture the subtle and sometimes conflicting views expressed within a single review. Nevertheless, sentiment analysis encompasses a variety of complex subproblems, and ABSA provides a clear methodology for addressing these challenges.

The field of ABSA has experienced significant advancements, primarily driven by deep learning models such as Transformers [2], which have reached state-of-the-art performance on academic benchmarks. However, this pursuit of enhanced performance has resulted in an increasing dependence on computationally intensive and opaque black-box models. For business practitioners, understanding the reasoning behind a model's predictions often holds more value than the predictions themselves. Without insight into how a model arrives at its conclusions, trust is diminished, and its effectiveness for data-driven decision-making is compromised.

This situation creates a critical tension, as highlighted by [3], who contends that for high-stakes decisions, the use of inherently interpretable models is not only preferable but essential. Instead of attempting to provide post-hoc explanations for opaque models, a practice that Rudin critiques as potentially misleading, the emphasis should be on developing models that are inherently transparent from the outset.

This paper advocates for re-centring the objective of ABSA in enterprise applications, shifting the emphasis from maximising predictive accuracy to generating actionable intelligence. To accomplish this, we introduce SP-ABSA[1], a framework designed not to surpass the benchmarks established by state-of-the-art generative models but to deliver solutions based on metrics that are critical to businesses: interpretability, traceability, and adaptability.

SP-ABSA is built on the linguistic principle that opinions are expressed through identifiable, recurring syntactic structures. Its primary aim is to extract sentiment triplets formatted as [Aspect, Attribute, Sentiment], for instance, *[battery life, long, positive]*. By grounding each extraction in clear grammatical rules, SP-ABSA ensures its outputs are transparent and reliable. This paper outlines the framework's methodology, its empirical foundations, and practical applications. Specifically, we formalise SP-ABSA as an interpretable framework that prioritises actionable insights over mere benchmark performance. A key component of this framework is a novel, unsupervised process for creating an aspect lexicon, enhancing the methodology's adaptability and efficiency. Our rule-based approach is validated through an empirical analysis of the syntactic patterns found in review corpora, reinforcing its data-driven basis. Furthermore, we present a clear argument for the business value of interpretable models while critically assessing the limitations of standard benchmarks in evaluating such systems.

The remainder of this paper is organised as follows: Section II reviews related work. Section III rationalises the empirical foundations for the core syntactic patterns. Section IV details the proposed methodology. Section V provides illustrative examples. Section VI presents our evaluation and results.

[1]The source code for our framework is publicly available at: https://github.com/rmansillal/Syntactic-Patterns-Aspect-Based-Sentiment-Analysis-SP-ABSA-

Section VII discusses the broader implications and limitations of our approach, and Section VIII concludes the paper.

II. Related Work

The field of ABSA is distinguished by two primary approaches: linguistic, rule-based systems and data-driven machine learning models. SP-ABSA is grounded in the former, providing a practical alternative to the latter.

A. Linguistic and Rule-Based Approaches

The foundation of rule-based ABSA is centred around linguistic structures. Early research conducted by Hu and Liu [4] employed part-of-speech (POS) tagging and frequency analysis to associate nouns (aspects) with nearby adjectives (opinions). This initial approach evolved into more sophisticated methods that utilised syntactic dependency parsing. A notable example is the Double Propagation (DP) algorithm [5], which uses a limited set of seed opinion words and handcrafted dependency rules to identify new aspects and opinion terms iteratively. While the DP algorithm is effective, its reliance on seed words makes it vulnerable to the initial lexicon and specific domain choices.

Further efforts have focused on automating the creation of these rules. Ruskanda et al. [6] implemented the Sequential Covering algorithm, enabling the automatic learning of dependency-based rules from data, thereby reducing manual effort. More recently, Mishra and Panda [7] formulated a comprehensive set of 30 rules based on a 'ROOT Node' technique. SP-ABSA builds upon this tradition by converting the output into structured triplets and, crucially, introducing an unsupervised lexicon creation process that enhances the analysis, making it more targeted and efficient compared to methods that require scanning every token or depend on predefined seed words.

B. The Machine Learning Paradigm

In contemporary research on ABSA, deep learning has emerged as the predominant paradigm. This transformation was largely driven by the introduction of attention-based Long Short-Term Memory (LSTMs) specifically tailored for ABSA [8]. Over time, it has progressed to the widespread implementation of large-scale Transformer models [9]–[11]. These models frequently redefine ABSA as a unified task, such as sequence-to-sequence generation. For instance, VQA-BERT [2] has achieved one of the highest F1 scores in the triplet extraction task, reaching approximately 0.579 on the same benchmark we use in our evaluation.

However, these models function within a fundamentally different framework. Their main objective is to optimise performance on specific benchmarks, with interpretability often relegated to a secondary concern. The complex, high-dimensional relationships these models learn are not easily converted into human-interpretable rules. This 'black-box' challenge constrains their utilisation in enterprise environments where traceability and accountability are essential. Consequently, SP-ABSA should not be considered as a direct competitor in terms of F1 score; rather, it is positioned as an alternative solution designed to meet different needs. In this respect, the value of a system is determined by its transparency and the actionable insights it provides. This perspective resonates with the critique posed by [3], who cautions that post-hoc explanations for black-box models often fail to accurately represent the model's underlying logic, potentially creating a misleading sense of security. The presence of this fidelity issue underscores the necessity for inherently transparent alternatives.

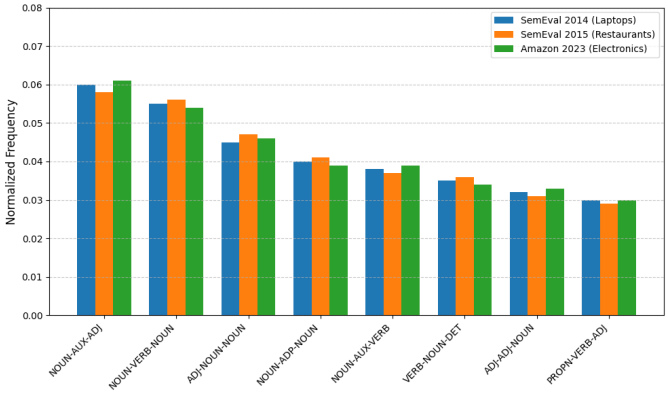

Fig. 1. Normalised frequency of the most common Part-of-Speech trigram patterns across three distinct review datasets.

III. Empirical Foundations: Prevalence of Syntactic Patterns

Before outlining the SP-ABSA methodology, it is vital to validate its core linguistic assumption: that opinions in user reviews are articulated through consistent and identifiable syntactic structures. To explore this, we conducted a thorough analysis of POS tag patterns across three distinct datasets: SemEval-2014 Task 4 (laptops) [12], SemEval-2015 Task 12 (restaurants) [13], and a large sample from the 2023 Amazon Reviews in the 'Electronics' category [14]. We extracted and quantified the frequency of POS trigram patterns to identify the most prevalent grammatical constructions.

The findings, illustrated in Fig. 1, reveal an evident consistency. Dominant patterns, such as Noun-Aux-Adj (e.g., 'screen is bright'), consistently rank among the most frequent. This empirical evidence lays a solid foundation for SP-ABSA, confirming that by focusing on these recurring patterns, our framework is tailored to capture the most common ways users express their opinions. This validation underscores that the syntactic structures targeted by our core patterns, including the Noun-Verb-Adjective construction detailed in Table I, which reflect the most frequent and dependable forms of opinion expression within this domain.

IV. Proposed Methodology: SP-ABSA framework

The SP-ABSA framework is structured as a two-stage architecture, as illustrated in Figure 2. The first stage involves an offline, corpus-level analysis aimed at generating a domain-specific aspect lexicon. This lexicon subsequently facilitates

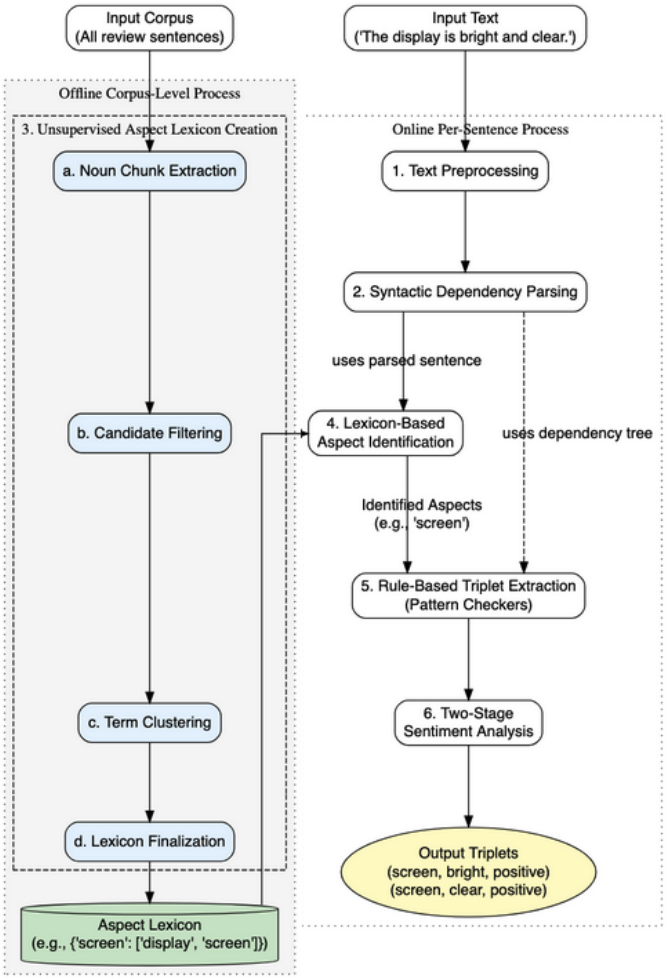

Fig. 2. The SP-ABSA framework, illustrating the two-stage process. The offline stage generates an aspect lexicon from the entire corpus, which then guides the online extraction of per-sentence triplets.

a highly efficient online pipeline for extracting [Aspect, Attribute, Sentiment] triplets from individual sentences. The architectural distinction between lexicon generation and triplet extraction is integral to the framework's adaptability and overall performance.

A. Stage 1: Offline Aspect Lexicon Generation

A fundamental component of our framework is the unsupervised generation of a domain-specific aspect lexicon derived from the raw input corpus. This process is conducted only once per dataset and serves as the semantic foundation for the online extraction pipeline. The procedure comprises four distinct steps, accompanied by detailed implementation instructions to ensure reproducibility.

Noun Chunk Extraction: We initiate the process by parsing the entire corpus using spaCy to identify all noun chunks. From each chunk, we extract a core aspect phrase by concatenating only those constituent tokens tagged as nouns (`NOUN`) or proper nouns (`PROPN`). This method effectively excludes determiners and adjectival modifiers within the chunk, thereby isolating the essential noun phrase (e.g., extracting 'screen' from 'the beautiful screen').

Candidate Filtering: The extracted phrases undergo a rigorous filtering process. We eliminate any phrase present in a standard stop word list or in a manually curated collection of generic domain terms (e.g., 'product', 'item', 'thing'). Additionally, phrases shorter than three characters are discarded. This step refines the candidate pool by removing semantically vacuous terms.

Term Clustering: To organise semantically related aspect phrases, we utilise a custom single-pass clustering procedure, as detailed in Algorithm 1. The algorithm initiates by processing candidate phrases sorted in descending order of frequency. For each phrase that has not yet been assigned to a cluster, the algorithm creates a new cluster. It then examines the remaining unclustered phrases, adding a phrase to the current cluster if it fulfils either of two criteria: a semantic similarity condition (where the cosine similarity of their GloVe vectors exceeds 0.8) or a morphological condition (where one phrase is a substring of the other). This hybrid approach efficiently groups terms like 'battery' and 'battery life,' which are related both semantically and structurally.

Algorithm 1 Unsupervised Aspect Term Clustering

Require: Candidate phrases $P = \{p_1, ..., p_n\}$, sorted by frequency
Require: Similarity threshold θ_{sim}
$Clusters \leftarrow \emptyset$, $Processed \leftarrow \emptyset$
for each phrase p_i in P **do**
  if $p_i \in Processed$ **then continue**
  end if
  $C \leftarrow \{p_i\}$
  for each phrase p_j in P where $j > i$ **do**
    if $p_j \in Processed$ **then continue**
    end if
    ▷ Hybrid similarity condition
    if AreSimilar(p_i, p_j, θ_{sim}) **then**
      $C \leftarrow C \cup \{p_j\}$
    end if
  end for
  $Clusters \leftarrow Clusters \cup \{C\}$
  $Processed \leftarrow Processed \cup C$
end for
return $Clusters$

function ARESIMILAR(p_i, p_j, θ_{sim})
  $v_i \leftarrow$ GetVector(p_i); $v_j \leftarrow$ GetVector(p_j)
  $is\_similar \leftarrow$ (hasVec$(v_i) \wedge$ hasVec$(v_j)) \wedge$ (CosSim$(v_i, v_j) > \theta_{sim}$)
  $is\_sub \leftarrow$ (isMultiWord$(p_i) \vee$ isMultiWord$(p_j)) \wedge$ $(p_i \subseteq p_j \vee p_j \subseteq p_i)$
  return $is\_similar$ **or** is_sub
end function

Lexicon Finalisation: Finally, for each resulting cluster, we designate the most frequently occurring term within that

cluster as the canonical aspect. All other terms in the cluster are then mapped to this canonical form, creating a many-to-one dictionary that constitutes the final aspect lexicon (e.g., 'screen': ['display', 'screen']). This curated lexicon represents the only output of the offline stage.

B. Stage 2: Online Triplet Extraction Pipeline

The online pipeline processes each sentence through a sequence of five stages to extract opinion triplets. Initially, the sentence undergoes standard text preprocessing, which includes converting to lowercase, removing URLs, standardising pronouns (e.g., changing 'it' to 'the product'), and normalising punctuation. Subsequently, the grammatical structure is analysed using the *en_core_web_lg* spaCy model, which generates a dependency tree and associated linguistic features. Once the sentence is parsed, the pre-computed lexicon is utilised for aspect identification. This identification occurs efficiently using spaCy's *PhraseMatcher*, which directly locates all occurrences of lexicon terms, thereby avoiding the costly linear scan of every token. Upon identifying an aspect, a set of rule-based *Pattern Checkers* is applied to the dependency tree to extract the corresponding attribute. These checkers encapsulate essential syntactic patterns, as detailed in Table I. For example, the Noun-Verb-Adjective rule identifies an adjectival complement (*acomp*) and links it to its nominal subject (*nsubj*). In the final stage, we assign a sentiment polarity to the extracted (Aspect, Attribute) pair through a transparent, rule-based model, ensuring full interpretability throughout the process. We employ a two-stage method that leverages NLTK's VADER [15], a lexicon and rule-based sentiment analysis tool.

During the first stage, we analyse the attribute in isolation, considering any syntactic negation. If the sentiment associated with the attribute is neutral or ambiguous, the second stage assesses the complete context of the sentence to reach a more informed polarity. This approach to sentiment analysis ensures that each component of the final triplet can be traced back to specific linguistic rules and verifiable lexicon entries.

TABLE I
CORE SYNTACTIC PATTERNS TARGETED BY SP-ABSA

Syntactic Structure	Example (Aspect-Attribute)	Key Dep[a]
Noun-Verb-Adjective	'The screen is bright.'	nsubj, acomp
Verb-Adjective-Noun	'It has a beautiful keyboard.'	attr, amod
Noun-Verb-Adverb	'Software works flawlessly.'	nsubj, advmod
Verbless Fragment	'Great screen.'	amod (on root)
Conjoined Attributes	'Screen is bright and clear.'	conj
Verb-Object Relation	'Update drains the battery.'	dobj, advmod

[a] Key syntactic dependencies from spaCy. **nsubj**: nominal subject; **acomp**: adjectival complement; **dobj**: direct object; **amod**: adjectival modifier; **attr**: attribute; **conj**: conjunct.

V. ILLUSTRATIVE EXAMPLES: THE VALUE OF INTERPRETABILITY

The following synthetic examples highlight SP-ABSA's capabilities and demonstrate why, in some contexts, its interpretable outputs are more valuable for decision-making than the opaque predictions of a 'black-box' model.

Example 1: Multiple Aspects and Negation

- **Input Sentence:** *'The keyboard is not comfortable and the trackpad isn't responsive.'*
- **SP-ABSA Output:**
  - (`keyboard`, `comfortable`, `negative`)
    Rule: `check_nva_acomp` + Negation
  - (`trackpad`, `responsive`, `negative`)
    Rule: `check_nva_acomp` + Negation

Within a business context, a product manager can clearly identify that two specific features are underperforming, and they can understand that this conclusion is drawn from a recognisable linguistic pattern. While a 'black-box' model may produce the same outcome, it does so without providing any underlying rationale, leaving the user to place blind trust in its results. Figure 3 visualises this parsing process, highlighting the essential negation dependency that accurately reverses the sentiment associated with the term 'comfortable.'

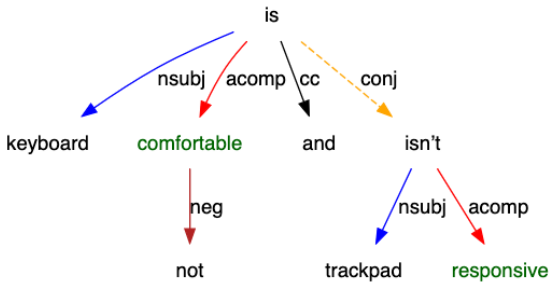

Fig. 3. Dependency Parse for Example 1.

Example 2: Conjoined Attributes

- **Input Sentence:** *'This is a fantastic gaming laptop; the screen is incredibly bright and clear.'*
- **SP-ABSA Output:**
  - (`gaming laptop`, `fantastic`, `positive`)
    Rule: `check_van_amod`
  - (`screen`, `bright`, `positive`)
    Rule: `check_nva_acomp`
  - (`screen`, `clear`, `positive`)
    Rule: `check_conjoined_attribute`

The system's capability to link both 'bright' and 'clear' to 'screen' through a specific conjunction rule provides a comprehensive understanding of their relationship. The connection to the rule verifies that the system has accurately interpreted the grammar. The dependency structure illustrated in Figure 4 demonstrates the conjunction relationship between 'bright' and 'clear.'

Example 3: Complex Verb-Object Relation

- **Input Sentence:** *'The new update drains the battery life too quickly.'*
- **SP-ABSA Output:**
  - (`battery life`, `quickly`, `negative`)
    Rule: `check_verb_dobj_advmod`

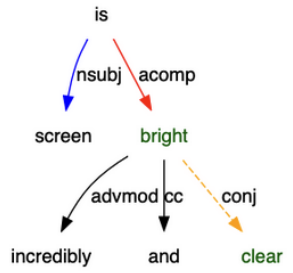

Fig. 4. Dependency Parse for Example 2.

Example 3 effectively highlights the distinctive value of a traceable, syntax-based framework. The output triplet, `(battery life, quickly, negative)`, when examined in isolation, may seem to lack context. However, the true insight is unveiled through its justification: Rule: `check_verb_dobj_advmod`.

This rule confirms that the system did not simply identify negative words near 'battery life.' Rather, it accurately parsed a complex grammatical structure where an *action* (the verb 'drains') is performed on an *object* (the aspect 'battery life'), with the manner of that action described by an *adverb* (the attribute 'quickly').

For analysts, this serves as an insightful diagnostic tool, demonstrating that the negative sentiment relates to a process rather than a static quality of the battery. The immediate follow-up question: 'What is causing this draining action?', can be addressed by inspecting the parse to determine the subject of the verb 'drains', which is 'the new update'. SP-ABSA conducts the linguistic analysis and provides a clear audit trail, enabling users to perform root-cause analysis with confidence. In contrast, an opaque model, despite producing the same triplet, would compel users to undertake this grammatical analysis manually, thereby severely restricting its 'actionable' nature.

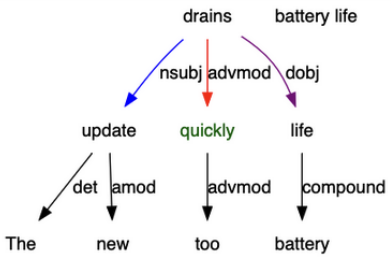

Fig. 5. Dependency Parse for Example 3.

VI. Evaluation and Results

While state-of-the-art generative models establish a high-performance benchmark (e.g., [2]), our aim is to showcase the clear and quantifiable improvements resulting from specific architectural enhancements within an interpretable, extractive framework. Crucially, this section argues that traditional quantitative metrics are insufficient for evaluating systems designed for actionable intelligence.

A. Dataset and Evaluation Metrics

We assessed SP-ABSA using the SemEval-2014 Task 4 dataset [12], a widely recognised benchmark for triplet extraction that includes annotated laptop reviews. Our evaluation approach consists of two main components. First, to illustrate the effectiveness of our framework within an interpretable paradigm, we directly compare SP-ABSA to a traditional dependency-based rule system, referred to as *Dep-Rule*. Second, to contextualise our findings and address the limitations of standard benchmarks, we compare our F1-score against the published results of a state-of-the-art generative model, VQA-BERT [2].

Our Dep-Rule baseline exemplifies a traditional dependency-rule approach, inspired by foundational work that first employed linguistic patterns to link aspects and opinions [4]. Specifically, this baseline serves as an implementation of systems that utilise syntactic dependency parsing to directly extract opinion targets, a methodology that has been applied in various forms [e.g., [16]]. In contrast to SP-ABSA, this system does not rely on a pre-computed aspect lexicon. Instead, it systematically analyses each sentence and applies two key syntactic patterns, adjectival modifiers and adjectival complements, to identify potential aspect-attribute pairs directly from the parse.

In evaluating the triplet extraction task, we utilise widely accepted performance metrics: Precision, Recall, and F1-Score [17]. To accommodate linguistic variability, we apply a partial matching criterion for aspect and attribute terms, requiring that one string be a substring of the other. In contrast, sentiment requires an exact match. A prediction is classified as a True Positive (TP) only if all three conditions are met.

B. Qualitative Error Analysis

Before evaluating quantitative scores, it is vital to first understand the nature of the model's outputs and the inherent limitations of standard benchmarks. Direct comparisons of F1-scores can be misleading when a model and a benchmark are optimised for different objectives. To address this, we conducted a qualitative analysis focused specifically on the False Positives (FPs) produced by our model, as these represent the primary point of divergence from the ground truth.

This necessary, manual, and time-intensive analysis involved a sample of 50 reviews that generated FP results. The investigation revealed a recurring category of extractions that, while penalised by the benchmark, constitute valid and actionable insights. As shown in Table II, SP-ABSA frequently identifies these 'correct but unannotated' triplets.

For instance, recognising that a laptop possesses 'many' features in addition to being 'great' is a valuable distinction for a product manager. Likewise, identifying the specific 'delete key' as the source of an editing problem provides a level of detail frequently absent from higher-level annotations. These

examples, uncovered during our sampled analysis, demonstrate that the true value of an interpretable system often lies in the rich, precise information that a rigid ground truth can overlook.

C. Performance Analysis

The quantitative results presented in Table III provide a clear depiction of the SP-ABSA framework's performance and the successful evolution of our methodology. Our refined SP-ABSA framework achieves a micro F1-score of 0.215, a relative improvement of over 38% compared to the Dep-Rule baseline F1-score of 0.155. This result empirically validates our lexicon-driven design.

The raw counts of False Positives (963) and False Negatives (1,164) provide these summary metrics with their essential context. The high FN count transparently reflects the non-exhaustive nature of our current rule set, providing a clear roadmap for future expansion. More importantly, the high FP count quantifies the scale of the misalignment between the discovery-oriented method of this study and the benchmark's fixed schema, an issue established in our qualitative analysis. As was demonstrated, a portion of these FPs are 'correct but unannotated' triplets, rather than model errors, which was confirmed through our manual analysis of a representative sample.

Consequently, the FP figure is conflated with an unknown number of valid, granular discoveries that are systematically penalised by the benchmark's ground truth. This context is crucial when considering the F1-score of 0.579 reported by state-of-the-art models, such as VQA-BERT [2]. The disparity highlights a fundamental limitation of using a single F1-score to assess a discovery-oriented system, as the metric cannot differentiate between model failure and valuable, granular insight.

D. Practical implications for SP-ABSA

The true value of SP-ABSA extends beyond its F1-score, it lies in its capacity to convert unstructured text into actionable intelligence. The structured triplets serve as high-quality raw materials for strategic visualisation.

Figure 6 exemplifies this transformation. By aggregating thousands of extracted triplets, the Priority Matrix provides a clear overview of the product landscape, plotting opinion volume against average sentiment. This empowers decision-makers to quickly identify strategic priorities, such as key strengths to 'Praise & Promote' (e.g., *screen*) as well as areas that require 'Urgent Action.'

While the matrix highlights *what* to focus on, SP-ABSA's interpretability clarifies *why*. Figure 7 offers this deeper insight. By examining a specific aspect, such as 'Screen,' we can visualise the particular, grammatically-grounded attributes contributing to its score. Each node represents an attribute extracted through a transparent rule, establishing a clear audit trail that traces back to the original text. This perspective reveals, for example, that the 'screen' is praised for its 'brightness' and 'vividness.' Such detailed, traceable insights are central to the interpretable paradigm, a capability that is fundamentally absent in opaque systems.

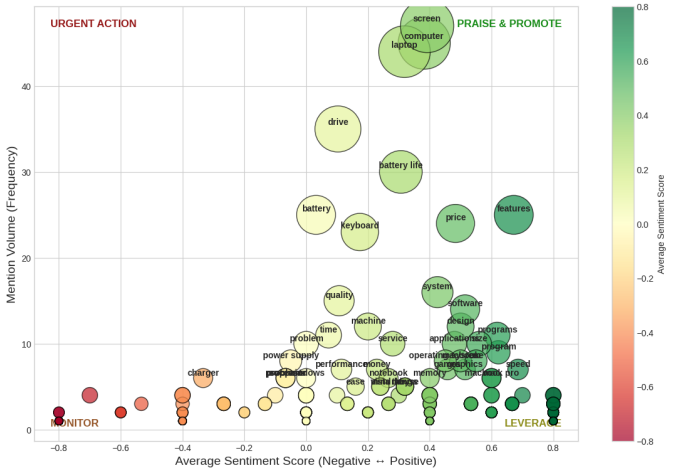

Fig. 6. A Priority Matrix visualisation derived from aggregated SP-ABSA triplet extractions. The x-axis represents the average sentiment score for each aspect, and the y-axis represents the mention volume (frequency) for each aspect.

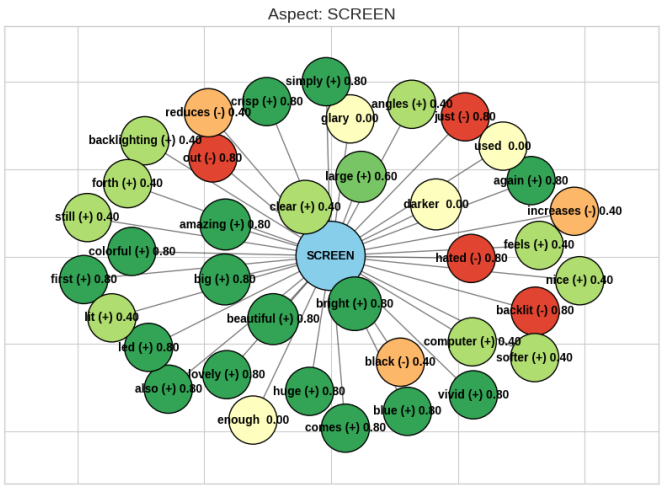

Fig. 7. Attribute network graph for the aspect 'screen'. The central node represents the canonical aspect, and each connected node corresponds to a specific attribute extracted by an SP-ABSA syntactic rule.

VII. Discussion

A. The Case for Interpretable ABSA in Practice

Our central view aligns with the increasing demand for trustworthy AI, particularly as highlighted by [3]. Rudin advocates for prioritising inherently interpretable models over opaque ones, especially in high-stakes decision-making. He argues that post-hoc explanations for black-box models often act as misleading approximations, creating a false sense of understanding. To tackle this challenge within the realm of business intelligence, we introduce SP-ABSA, which is designed to provide transparent outputs through clearly defined syntactic rules, making the model self-explanatory.

Our findings offer compelling empirical support for this viewpoint. The notable gap between SP-ABSA's F1-score (0.215) and its demonstrated ability to extract 'correct but

TABLE II
QUALITATIVE EXAMPLES WHERE SP-ABSA IDENTIFIES VALID, 'CORRECT BUT UNANNOTATED' TRIPLETS MISSED BY THE SEMEVAL GROUND TRUTH. NOVEL DETECTIONS ARE SHOWN IN BOLD.

Customer Review	**Ground Truth**	**SP-ABSA Output**
Great laptop that offers many great features!	`[('features', 'great', 'positive')]`	`[('features', 'many', 'positive'), ('features', 'great', 'positive')]`
And if you do a lot of writing, editing is a problem since there is no forward delete key.	`[('editing', 'problem', 'negative')]`	`[('key', 'delete', 'negative'), ('editing', 'problem', 'negative')]`
...I am operating an incredibly efficient and useful machine for a great price.	`[('price', 'great', 'positive')]`	`[('machine', 'efficient', 'positive'), ('price', 'great', 'positive')]`
Awesome laptop and the perfect size to carry around in college.	`[('size', 'perfect', 'positive')]`	`[('laptop', 'awesome', 'positive'), ('size', 'perfect', 'positive')]`
It is in the best condition and has a really high quality.	`[('quality', 'high', 'positive')]`	`[('condition', 'best', 'positive'), ('quality', 'high', 'positive')]`
Overall, this laptop is definitely a keeper with its simple yet stylish design...	`[('design', 'stylish', 'positive')]`	`[('laptop', 'keeper', 'positive'), ('design', 'stylish', 'positive')]`
Besides the great look, it is a great machine.	`[('look', 'great', 'positive')]`	`[('machine', 'great', 'positive'), ('look', 'great', 'positive')]`

TABLE III
QUANTITATIVE PERFORMANCE OF INTERPRETABLE MODELS

Model	**TP**	**FP**	**FN**	**Prec.**	**Rec.**	**F1**
Dep-Rule Baseline	212	1076	1244	0.165	0.146	0.155
SP-ABSA (Ours)	**292**	**963**	**1164**	**0.233**	**0.201**	**0.215**

For Context: SOTA (VQA-BERT) F1 is 0.579 [2].* *Our work does not aim to compete on these metrics but rather to critique their limitations.

unannotated' insights (as presented in Table II) emphasises a misalignment between traditional academic benchmarks and the requirements of enterprise applications. These benchmarks are structured to reward models that excel at pattern recognition against a fixed ground truth. However, this methodology may overlook the verifiable and actionable intelligence that a more nuanced, linguistically grounded system can provide.

Consequently, this study advocates for a dual-axis evaluation approach for these systems. While standard metrics offer critical contextual insight, they should be supplemented by qualitative analyses that assess a system's utility based on the transparency and actionability of its outputs. Future research should not only aim to advance interpretable models but also to develop evaluation methodologies that enable the discovery of novel, verifiable insights, rather than penalising deviations from an incomplete ground truth.

B. Methodological Implications

The divergence between the outputs of SP-ABSA and the benchmark's ground truth arises from a fundamental methodological difference: the SP-ABSA framework utilises an unsupervised, bottom-up approach for discovery, whereas state-of-the-art generative models are predominantly trained using a supervised, top-down method for recognition.

Generative models, such as those trained on datasets like SemEval, are designed to recognise and map text to a predefined annotation schema, often favouring canonical, high-level aspect terms (e.g., 'price,' 'quality'). However, this supervised methodology can limit the identification of more specific or novel aspects that fall outside this schema. For instance, a model heavily trained on the abstract aspect 'editing' might fail to recognise the actual root cause, the 'delete key', as a distinct aspect.

In contrast, SP-ABSA produces its understanding of aspects from the linguistic evidence within the corpus itself. Its approach to unsupervised lexicon creation is not bound by a predefined schema, enabling it to discover the specific, detailed terms that users actually employ. Furthermore, benchmark annotations frequently emphasise the single most salient opinion. For example, in the review 'Great laptop that offers many great features!', an annotator might capture only the primary opinion as ('features,' 'great,' 'positive'), overlooking the grammatically accurate yet secondary quantifier 'many.' As a linguistic engine, SP-ABSA extracts both the primary and secondary opinions, yielding a more precise interpretation of the text. This technical distinction is the primary reason why direct F1-score comparisons can be misleading, as they often penalise the framework for this distinction.

C. Limitations and Future Directions

A significant strength of SP-ABSA is its end-to-end interpretability, achieved through the integration of the transparent VADER sentiment model. However, this design choice also defines clear avenues for future research. While robust, VADER's general-purpose lexicon may not fully capture domain-specific sentiment nuances (e.g., 'heavy' is negative for a laptop but potentially positive for a camera tripod). A promising direction is to develop a method for augmenting VADER with domain-specific sentiment terms, potentially sourced from the corpus itself, thereby creating a more tailored and accurate, yet interpretable, model.

Furthermore, our quantitative results, particularly the raw counts of False Positives (963) and False Negatives (1164), provide a clear roadmap for improving the framework's linguistic coverage. The preliminary analysis of a representative sample of FPs, which was instrumental in identifying the 'correct but unannotated' triplets in Table II, underscores the value of this approach. A key priority for future work is to conduct a comprehensive and systematic analysis of all False Positives and False Negatives. This deep dive will allow us to identify and categorise overlooked grammatical structures, such as those in comparative sentences or conditional clauses. By encoding these patterns as new, modular checker functions, we aim to strategically enhance our model's recall and precision while preserving the transparency that defines SP-ABSA.

VIII. Conclusion

This paper presented SP-ABSA, a lexicon-driven framework for ABSA that prioritises interpretability and actionable intelligence over raw benchmark performance. This study outlined its two-stage architecture and demonstrated its superior performance compared to a baseline interpretable system. More importantly, we advocated for a reassessment of how such systems are evaluated within enterprise contexts.

By employing qualitative evidence to highlight the shortcomings of standard benchmarks, SP-ABSA challenges the belief that a higher F1-score necessarily equates to greater business value. It provides a methodology for converting raw customer feedback into a strategic asset, ensuring that each insight is founded on a clear and auditable linguistic rationale. Ultimately, this work emphasises that in the pursuit of actionable intelligence, the most valuable insights are not just predicted but thoroughly understood.

References

[1] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE intelligent systems*, vol. 32, no. 6, pp. 74–80, 2018.

[2] H. Yan, J. Dai, X. Qiu, Z. Zhang, Y. Geng, and B. Li, "A unified generative framework for aspect-based sentiment analysis," arXiv preprint arXiv:2106.04300, 2021.

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[4] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2004, pp. 168–177.

[5] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Computational Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[6] F. Z. Ruskanda, D. H. Widyantoro, and A. Purwarianti, "Sequential covering rule learning for language rule-based aspect extraction," in *2019 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 2019, pp. 229–234.

[7] P. Mishra and S. K. Panda, "Dependency structure-based rules using root node technique for explicit aspect extraction from online reviews," *IEEE Access*, vol. 11, pp. 65 117–65 137, 2023.

[8] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, 2019, pp. 4171–4186.

[10] H. Fei, Y. Ren, Y. Zhang, and D. Ji, "Nonautoregressive encoder–decoder neural framework for end-to-end aspect-based sentiment triplet extraction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5544–5556, 2021.

[11] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 019–11 038, 2022.

[12] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, P. Nakov and T. Zesch, Eds. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 27–35. [Online]. Available: https://aclanthology.org/S14-2004/

[13] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, P. Nakov, T. Zesch, D. Cer, and D. Jurgens, Eds. Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 486–495. [Online]. Available: https://aclanthology.org/S15-2082/

[14] Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and J. McAuley, "Bridging language and items for retrieval and recommendation," *arXiv preprint arXiv:2403.03952*, 2024.

[15] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, 2014, pp. 216–225.

[16] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, 2014, pp. 28–37.

[17] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.