Applied Natural Language Processing 2023 S1 Assignment 3: Building an aspect-based sentiment analysis algorithm based on syntactic parsing

Akide Liu
STUDENT NO. 1743748

June 23, 2023

Report submitted for **COMP SCI 4817** at the Faculty of Sciences, Engineering and Technology, University of Adelaide

In submitting this work I am indicating that I have read the University's Academic Integrity Policy. I declare that all material in this assessment is my own work except where there is clear acknowledgement and reference to the work of others.

I give permission for this work to be reproduced and submitted to other academic staff for educational purposes.

In this report, first-person pronouns such as "WE" or "I" have been used interchangeably to refer to the **author's own work**, in order to maintain a professional and academic tone.

OPTIONAL: I give permission this work to be reproduced and provided to future students as an exemplar report.

Contents

1 Introduction to Aspect-Based Sentiment Analysis

Aspect-Based Sentiment Analysis (ABSA) [4] is an essential sub-discipline within the broader field of sentiment analysis [6] . It primarily focuses on extracting and analyzing sentiments that are expressed towards particular aspects or attributes of a given target entity. ABSA finds its utility in a plethora of domains such as product reviews, social media discussions, customer feedback, etc. It is vital for comprehending the subtleties in the opinions and attitudes expressed towards different aspects of a target entity. [10]

The fundamental objective of ABSA is to pinpoint the sentiment polarity (positive, negative, or neutral) that is associated with each aspect mentioned in a given text [8] . The term *aspect* can pertain to distinctive features, components, or attributes of a product, service, or any entity. For instance, in the context of a restaurant review, aspects could encompass the quality of food, service, ambiance, and price. ABSA encompasses multiple stages.

1. **Aspect Extraction** [9, 5] : Initially, the relevant aspects present in the text need to be identified and extracted. This involves processing the text data and pinpointing the key elements that are indicative of aspects.

2. **Sentiment Classification** [9, 14] : Subsequently, sentiment classification is undertaken to ascertain the sentiment polarity tied to each extracted aspect. Typically, this stage is underpinned by machine learning methodologies that utilize annotated datasets to train models capable of accurately classifying sentiment.

3. **Integration**: Lastly,[9, 13] the outcomes of aspect extraction and sentiment classification are amalgamated to render a comprehensive aspect-level sentiment analysis of the text.

ABSA has garnered considerable attention owing to its extensive applicability across different sectors. It equips businesses with the capability to garner insights into consumer opinions, pinpoint areas necessitating improvement, and facilitate data-driven decision-making. Moreover, ABSA is instrumental in monitoring brand reputation, deciphering consumer preferences, and enabling targeted marketing strategies. [3]

Research in ABSA is a fast-evolving domain [7, 15, 12] . Scholars and practitioners are incessantly exploring diverse techniques and methodologies to augment aspect extraction, sentiment classification, and enhance the overall efficacy of ABSA. Some of the approaches that have been explored include rule-based methods, machine learning algorithms, deep learning models, and hybrid methods that amalgamate different techniques to tackle the intricacies of ABSA. ABSA typically entails three cardinal tasks:

- **Aspect Term Extraction**: This refers to the identification of specific aspects or attributes discussed in the text. Aspect terms can be either explicit, such as "food" and "service" in a restaurant review, or implicit.

- **Aspect Category Detection**: Post the extraction of aspect terms, they are categorized into predefined groups. For instance, the term "food" could be assigned to the category "menu", while "service" may be classified under "staff".

- **Sentiment Polarity Detection**: This phase involves determining the sentiment polarity, be it positive, negative, or neutral, of each identified aspect.

Aspect-Based Sentiment Analysis is a powerful tool that allows for a nuanced understanding of textual data. By focusing on specific aspects within the text and analyzing the sentiment associated with these aspects, ABSA provides more granular insights than traditional sentiment analysis. This has a wide range of applications, from helping businesses understand customer feedback to aiding in data-driven decision-making. The continual evolution of methodologies in ABSA holds promise for even more sophisticated analysis capabilities in the future.

Clausal Argument Relations	
Abbreviation	**Description**
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	
Abbreviation	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	
Abbreviation	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Table 1: Dependency Relations in Natural Language Processing

1.1 Dependency Relations in ABSA

Dependency relations serve as a guide to how different words within a sentence are connected to each other. They can be utilized to precisely extract aspects and understand the sentiment expressed about them. The following are some of the key dependency relations from the table above that play a crucial role in ABSA as shown in table 1 :

- **NSUBJ (Nominal Subject)**: Identifying the nominal subject in sentences is essential to find out what the sentence is mainly talking about, which often turns out to be an aspect.

  vbnet Copy code

- **DOBJ (Direct Object)**: Often the opinion in the sentence is directed towards the direct object, making it another potential aspect.

- **AMOD (Adjectival Modifier)**: This relation is crucial for linking adjectives to the nouns they modify, which is often indicative of sentiment. For example, in delicious pizza, delicious is an adjectival modifier of pizza.
- **NMOD (Nominal Modifier)**: Similar to adjectival modifiers, nominal modifiers modify a noun and can be indicative of sentiment or help in aspect extraction.
- **DET (Determiner)** and **CASE (Case Marking)**: These relations can provide context to an aspect. For example, in the phrase ”the battery life of the phone”, the determiner ”the” and case marking ”of” help in identifying ”battery life” as the aspect.
- **CONJ (Conjunct)**: Conjuncts can link aspects or sentiments. For example, in ”the screen is clear and bright”, the conjunct ”and” links two sentiments about the aspect ”screen”.

There are also some examples in table 2

Relation	Examples with *head* and **dependent**
NSUBJ	**United** *canceled* the flight.
DOBJ	United *diverted* the **flight** to Reno.
	We *booked* her the **first** flight to Miami.
IOBJ	We *booked* **her** the flight to Miami.
NMOD	We took the **morning** *flight*.
AMOD	Book the **cheapest** *flight*.
NUMMOD	Before the storm JetBlue canceled **1000** *flights*.
APPOS	*United*, a **unit** of UAL, matched the fares.
DET	**The** *flight* was canceled.
	Which *flight* was delayed?
CONJ	We *flew* to Denver and **drove** to Steamboat.
CC	We flew to Denver **and** *drove* to Steamboat.
CASE	Book the flight **through** *Houston*.

Table 2: Examples of Dependency Relations in Sentences

1.2 Parsing Example

```
'id': '3121',
'text': 'But the staff was so horrible to us.',
'aspect_terms': [{'term': 'staff','polarity': 'negative','from': '8','to': '13'}],
'aspect_categories': [{'category': 'service', 'polarity': 'negative'}],
'words': ['But', 'the', 'staff', 'was', 'so', 'horrible', 'to', 'us.'],
'mask': [False, False, True, False, False, False, False, False]
```

Figure 1: Example 1 of row data format.

In the given example (fig. 1), the input text is ”But the staff was so horrible to us.” accompanied by the aspect term *staff*, which is associated with a **negative** polarity. By parsing the input sentence into a dependency tree as depicted in fig. 2, a more granular inspection of the aspect-based sentiment through affiliated terms can be conducted.

To evaluate the sentiment with respect to the aspect term *staff*, a traversal through the dependency tree can be initiated from the token representing *staff*. Observing the dependency tree, it is evident that "staff" functions as a child node of "was" through a *nsubj* (nominal subject) dependency relation. Subsequently, the tree traversal root is repositioned to "was".

In the next step of the traversal, it is identified that the token "horrible," which is of type ADJ (adjective), is a child node of the tree root ("was") connected via an *acomp* (adjectival complement) relation. This observation can be formalized into a rule for the automated extraction of sentiment-related tokens:

1. Set the current element to the head of the aspect term token.

$$\text{element} = \text{term.head} \quad \Rightarrow \quad \text{"was"} = \text{"staff".head}$$

2. Traverse through the children of the current tree root.

$$\text{for } c \text{ in element.children}$$

3. If a child node has a dependency type "acomp," submit the token for sentiment analysis to ascertain whether the sentiment is positive, negative, or neutral.

$$\text{if } c.dep\_ == \text{"acomp" then submit } c \text{ for sentiment analysis}$$

This systematic approach harnesses the dependency relations in the sentence to effectively identify and analyze the sentiment expressed towards a particular aspect. The methodology enables a nuanced understanding of sentiment and can be incorporated into Aspect-Based Sentiment Analysis systems for robust analysis of textual data.

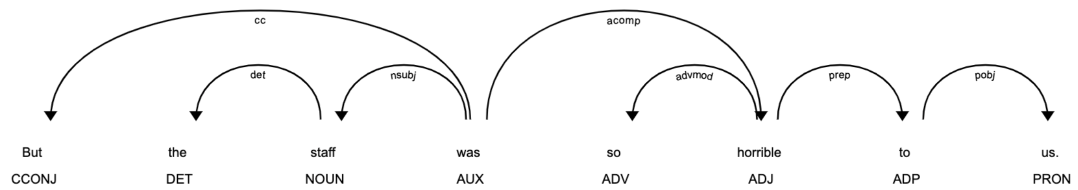


Figure 2: Example 1 of dependency tree of a sentence.

2 Data Preparation

2.1 Dataset Description

The restaurant dataset under consideration is a benchmark corpus extensively utilized for evaluating the efficacy of Aspect-Based Sentiment Analysis (ABSA) algorithms. This dataset is consolidated within a single XML file and encompasses 3041 records, each comprising multiple attributes, namely: *id*, *text*, *aspect_terms*, and *aspect_categories*.

The *aspect_terms* attribute contains what is referred to as the aspect term, which essentially is a keyword employed as an initializing query for parsing. It is important to note that these aspect terms serve as focal points in the ABSA, being the entities towards which sentiments are expressed within the text.

Additionally, sentiments expressed toward these aspect terms are categorized into four distinct classes: positive, negative, neutral, and conflict. The term 'conflict' here denotes instances where mixed sentiments are expressed towards a singular aspect term.

This study is predominantly centered on the utilization of aspect terms for implementing rule-based, aspect-oriented sentiment analysis. The rule-based methodology enables the creation of algorithms that systematically identify and analyze sentiments by adhering to a predefined set of rules, often incorporating linguistic patterns and structures. Such an approach is particularly beneficial in scenarios where machine learning models may not be feasible due to constraints in data or computational resources. Through this method, we aim to discern patterns and relations among aspect terms and the sentiments attributed to them, which in turn paves the way for a deeper understanding and finer analysis of the opinions embedded in the dataset.

Algorithm 1 Parse XML Data to Data Structure

```
function PARSE_XML_TO_DF(root)
    parsed_data ← empty list
    for sentence in root.findall('sentence') do
        sentence_data ← dictionary with keys: 'id', 'text', 'aspect_terms', 'aspect_categories'
        sentence_data['id'] ← sentence.get('id')
        sentence_data['text'] ← sentence.find('text').text
        aspect_terms ← sentence.find('aspectTerms')
        if aspect_terms is not None then
            for term in aspect_terms.findall('aspectTerm') do
                term_details ← dictionary with keys: 'term', 'polarity', 'from', 'to'
                term_details['term'] ← term.get('term')
                term_details['polarity'] ← term.get('polarity')
                term_details['from'] ← term.get('from')
                term_details['to'] ← term.get('to')
                append term_details to sentence_data['aspect_terms']
            end for
        end if
        aspect_categories ← sentence.find('aspectCategories')
        if aspect_categories is not None then
            for category in aspect_categories.findall('aspectCategory') do
                category_details ← dictionary with keys: 'category', 'polarity'
                category_details['category'] ← category.get('category')
                category_details['polarity'] ← category.get('polarity')
                append category_details to sentence_data['aspect_categories']
            end for
        end if
        append sentence_data to parsed_data
    end for
    return parsed_data
end function
```

2.2 XML Parsing and Data Extraction

In this section, we elaborate on the operations involved in parsing XML data and extracting relevant information for Aspect-Based Sentiment Analysis (ABSA). Given that the dataset is structured in XML format, it is essential to develop a mechanism to parse this data into a more accessible data structure for further processing and analysis.

2.2.1 Parsing Algorithm Description

The algorithm shows in algorithm 1 employed for parsing XML data into a suitable data structure is named `parse_xml_to_df`. This function accepts an XML root element and iterates through each sentence element contained within the root.

For each sentence, the function extracts essential attributes such as sentence ID and text. In addition to this basic information, the algorithm extracts aspect terms and aspect categories, which are critical to ABSA. Each sentence may contain multiple aspect terms and categories.

- For each aspect term within a sentence, the algorithm extracts the term itself, the sentiment polarity, and the positions in the text where the term occurs.
- For each aspect category within a sentence, the algorithm extracts the category and its sentiment polarity.

All the extracted information is aggregated into a dictionary with the keys: `id`, `text`, `aspect_terms`, and `aspect_categories`. The `aspect_terms` and `aspect_categories` are lists of dictionaries containing information regarding each aspect term and category, respectively.

The dictionaries corresponding to each sentence are then appended to a list, which will ultimately contain the information for all sentences. This list is the final output of the function.

2.2.2 Output Data Structure

The output of the `parse_xml_to_df` function is a list, where each element is a dictionary representing an individual sentence from the XML data. Each dictionary contains the following keys:

- `id`: A unique identifier corresponding to the sentence.
- `text`: The textual content of the sentence.
- `aspect_terms`: A list of dictionaries, where each dictionary represents an aspect term and contains the keys:
  - `term`: The aspect term itself.
  - `polarity`: The sentiment polarity associated with the aspect term.
  - `from`: The starting position of the aspect term in the text.
  - `to`: The ending position of the aspect term in the text.
- `aspect_categories`: A list of dictionaries, where each dictionary represents an aspect category and contains the keys:

- `category`: The category itself.
- `polarity`: The sentiment polarity associated with the category.

This output data structure serves as the foundation for the subsequent steps in Aspect-Based Sentiment Analysis, facilitating efficient processing and analysis of the extracted information.

3 Rule Oriented Aspect based Sentiment System

3.1 Overall System

The objective of this research is to design and implement a Rule-Oriented Aspect-Based Sentiment Analysis System (ROAB-SAS) that is lightweight, efficient, and structurally coherent. The system's purpose is to process textual input sentences and analyze the sentiment polarity with respect to the provided aspects. The core strength of this system lies in its modular architecture, which is engineered through decomposition. The algorithmic logic and rule definitions are segregated into distinct sections, thereby enhancing the systems extensibility. This modular approach facilitates easy integration of additional rules and enables rule chaining for each aspect term.

Algorithm 2 Aspect Sentiment Analyzer

procedure ASPECTSENTIMENTANALYZER(sentences, aspect_terms, debug=False)
  $nlp \leftarrow$ load_spacy_model("en_core_web_sm")
  $poswords \leftarrow$ opinion_lexicon.positive()
  $negwords \leftarrow$ opinion_lexicon.negative()
  $sentences \leftarrow sentences$
  $aspect\_terms \leftarrow aspect\_terms$
  $debug \leftarrow debug$
  $sentiment\_map \leftarrow \{positive : 1, negative : -1, neutral : 0\}$
  $new\_aspect\_term\_state \leftarrow \{\}$
end procedure

In the context of this system, the initial step involves the definition of a class known as 'AspectSentimentAnalyzer.' This class assumes the responsibility of managing the global states of the system. It undertakes the crucial task of initializing fundamental components, which include Natural Language Processing (NLP) tools, aspect terms, and sentiment lexicons show in algorithm 2.

Subsequently, a code function is implemented, serving as the main point of invocation for conducting aspect-based sentiment analysis. This function is invoked whenever there is a need to analyze the sentiment associated with specific aspects shows in algorithm 3.

To ensure a systematic and comprehensive analysis, a chain of rules is applied during the sentiment analysis process. These rules are executed sequentially, one after another, to perform various tasks related to aspect-based sentiment evaluation shows in algorithm 4.

Furthermore, the system incorporates an early-stage conditional checking mechanism, which aims to identify and handle cases where mismatches occur. This tool enables the system to proactively identify and address any inconsistencies or discrepancies that may arise during the analysis, ensuring the accuracy and reliability of the results shows in algorithm 5.

Algorithm 3 Analysis Endpoint

```
procedure ANALYZE
    doc ← nlp(sentences)
    aspect_terms_tokens ← [term.term.lower() for term in aspect_terms]    ▷ Modify aspect terms
    for each ast in aspect_terms_tokens do
        if ' ' in ast then
            aspect_terms_tokens ← ast.split()
        end if
    end for
    apply_rules(doc)
    return new_aspect_term_state
end procedure
```

Algorithm 4 Rule Chain

```
procedure APPLY_RULES(doc)
    rule_list ← [rule1, rule2, ..., rule11]
    number_of_rules ← min(number_of_rules, len(rule_list))
    for each token in doc do
        if token.text.lower() in aspect_terms_tokens then
            new_aspect_term_state[token.text] ← 0
            for each rule in rule_list do
                rule(token)
            end for
        end if
    end for
end procedure
```

Algorithm 5 Confidential Assertion

```
procedure CHECK_CORRECTNESS(state)
    state ← state.copy()
    for each (s, v) in state.items() do
        if abs(v) > 0 then
            state[s] ← 1 if v > 0 else −1
        end if
    end for
    for each term in aspect_terms do
        if term['term'] has None then Continue
        end if
        if state[term['term']] ≠ sentiment_map[term['polarity']] then
            print(state, aspect_terms)
            return False
        end if
    end for
    return True
end procedure
```

3.2 Rules

3.2.1 Rule 1: Term's children with "amod" dependency relation

1. If the aspect term's child is with dependency type "amod", if a positive score is obtained by *TextBlob*, then the sentiment towards the aspect term is positive.

2. If the aspect term's child is with dependency type "amod", if a negative score is obtained by *TextBlob*, then the sentiment towards the aspect term is negative.

3. If the aspect term's child is with dependency type "amod", if a zero score is obtained by *TextBlob*, then the sentiment towards the aspect term is neutral.

These rules are derived from the example provided in the assignment files. The dependency type "amod" refers to an adjectival modifier, which can represent the sentiment associated with the term. Therefore, if a word has an "amod" tag and is not found in the negative or positive words list, it is considered to have a neutral sentiment.

3.2.2 Rule 2: Term's head's children with "acomp" dependency relation

1. If the sentiment score obtained for the aspect term's head's children is positive according to the *TextBlob* analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head's children is negative according to the *TextBlob* analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head's children is zero according to the *TextBlob* analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term's head has children with a dependency type of "acomp," the sentiment analysis using *TextBlob* is used to determine the sentiment towards the aspect term. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.3 Rule 3: Term's Children with "acomp" or "amod" Dependency Relation and "ADJ" Attribute

1. If the sentiment score obtained for the aspect term's children with "acomp" or "amod" dependency relation and "ADJ" attribute is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's children with "acomp" or "amod" dependency relation and "ADJ" attribute is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's children with "acomp" or "amod" dependency relation and "ADJ" attribute is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term has children with a dependency type of "acomp" or "amod" and an attribute of "ADJ," a sentiment analysis tool is utilized to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's children. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.4 Rule 4: Term's Head's Children with "advmod" Dependency Relation and "ADV" Attribute while Head is a "VERB"

1. If the sentiment score obtained for the aspect term's head's children with a "advmod" dependency relation and "ADV" attribute, while the head is a "VERB," is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head's children with a "advmod" dependency relation and "ADV" attribute, while the head is a "VERB," is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head's children with a "advmod" dependency relation and "ADV" attribute, while the head is a "VERB," is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term has children with a dependency type of "advmod" and an attribute of "ADV," and the head of the aspect term is a "VERB," a sentiment analysis tool is employed to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's head's children. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.5 Rule 5: Term's Head's Children with "advmod" Dependency Relation and "ADV" Attribute while Head is a "VERB", recessively.

This rule is an enhancement of Rule 4, we add additional reclusive processing.

3.2.6 Rule 6: Term's Head's Children with "compound" Dependency Relation and "NOUN" Attribute

1. If the sentiment score obtained for the aspect term's head's children with a "compound" dependency relation and "NOUN" attribute is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head's children with a "compound" dependency relation and "NOUN" attribute is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head's children with a "compound" dependency relation and "NOUN" attribute is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term has children with a dependency type of "compound" and an attribute of "NOUN," a sentiment analysis tool is utilized to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's head's children. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.7 Rule 7: Term's Head's Head's Children with "dobj" Dependency Relation

1. If the sentiment score obtained for the aspect term's head's head's children with a "dobj" dependency relation is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head's head's children with a "dobj" dependency relation is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head's head's children with a "dobj" dependency relation is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term has children with a dependency type of "dobj" in relation to the aspect term's head's head, a sentiment analysis tool is employed to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's head's head's children. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.8 Rule 8: Term's Head's Children with "neg" Dependency Relation

1. The presence of a child with a "neg" dependency relation indicates a negation of the sentiment towards the aspect term.

2. Therefore, regardless of other sentiment analysis scores, the sentiment towards the aspect term is considered negative if a child with a "neg" dependency relation is present.

This rule suggests that when the aspect term has children with a dependency type of "neg" in relation to the aspect term's head, the sentiment towards the aspect term is automatically considered negative. The presence of a child with a "neg" dependency relation indicates a negation of the sentiment towards the aspect term. Thus, regardless of other sentiment analysis scores, if a child with a "neg" dependency relation is present, the sentiment towards the aspect term is considered negative.

3.2.9 Rule 9: Term's Head's Children with "acomp" and not "neg" Dependency Relation

1. If the sentiment score obtained for the aspect term's head's children with a "acomp" dependency relation and without a "neg" dependency relation is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head's children with a "acomp" dependency relation and without a "neg" dependency relation is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head's children with a "acomp" dependency relation and without a "neg" dependency relation is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term has children with a dependency type of "acomp" in relation to the aspect term's head, and there are no children with a "neg" dependency relation, a sentiment analysis tool is utilized to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's head's children with a "acomp" dependency relation but without a "neg" dependency relation. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.10 Rule 10: Term's Head has "NOUN" Attribute

1. If the sentiment score obtained for the aspect term's head with a "NOUN" attribute is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head with a "NOUN" attribute is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head with a "NOUN" attribute is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term's head has an attribute of "NOUN," a sentiment analysis tool is employed to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's head. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

3.2.11 Rule 11: Term's Head's Head's Children with "advmod" Dependency Relations

1. If the sentiment score obtained for the aspect term's head's head's children with "advmod" dependency relations is positive according to the sentiment analysis, then the sentiment towards the aspect term is considered positive.

2. If the sentiment score obtained for the aspect term's head's head's children with "advmod" dependency relations is negative according to the sentiment analysis, then the sentiment towards the aspect term is considered negative.

3. If the sentiment score obtained for the aspect term's head's head's children with "advmod" dependency relations is zero according to the sentiment analysis, then the sentiment towards the aspect term is considered neutral.

This rule suggests that when the aspect term has children with a dependency type of "advmod" in relation to the aspect term's head's head, a sentiment analysis tool is employed to determine the sentiment towards the aspect term. The sentiment analysis is based on the sentiment scores obtained from the analysis of the aspect term's head's head's children with "advmod" dependency relations. Positive, negative, or neutral sentiments are assigned based on the sentiment score obtained from the analysis.

4 Evaluation

4.1 Evaluation Metrics

The performance of a classification model can be assessed using various evaluation metrics, including, precision, recall, and F1 score. These metrics are defined as follows:

1. Precision: The proportion of true positive instances among those predicted as positive.
$$\text{Rrecision} \ = \frac{tp}{tp + fp} \tag{1}$$

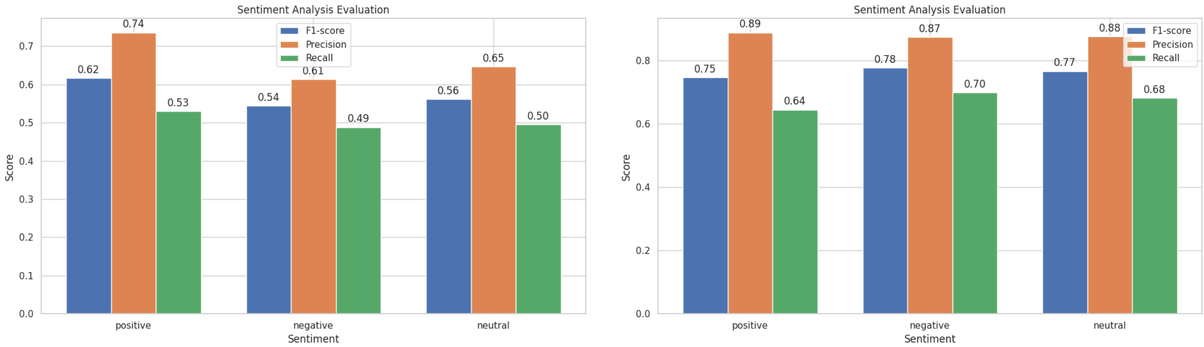
2. Recall (Sensitivity): The proportion of true positive instances among the actual positive instances.
$$\text{Recall} \ = \frac{tp}{tp + fn} \tag{2}$$

3. F1 Score: The harmonic mean of precision and recall, used as a balanced measure of both metrics.
$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

4.2 Performance Comparison

Table 3: Aspect based Sentiment Analysis Evaluation

Method	Sentiment	Precision	Recall	F1-Score
Our Rule	Positive	0.7360	0.5304	0.6165
	Negative	0.6139	0.4885	0.5441
	Neutral	0.6465	0.4957	0.5612
Sentiment Bert [1]	Positive	0.8886	0.6448	0.7473
	Negative	0.8738	0.6995	0.7770
	Neutral	0.8763	0.6819	0.7670



(a) Our Rule (b) Sentiment Bert

Figure 3: Aspect based Sentiment Analysis Evaluation

The table presents the results of sentiment analysis evaluation using two different methods: "Our Rule" and "Sentiment Bert". The evaluation metrics include Precision, Recall, and F1-Score, which are measures commonly used to assess the performance of sentiment analysis models.

In terms of Precision, "Sentiment Bert" outperforms "Our Rule" for all sentiment categories. It achieves a higher Precision score for Positive, Negative, and Neutral sentiments, indicating that it has a higher accuracy in correctly classifying the sentiments.Regarding Recall, "Sentiment Bert" also shows better performance compared to "Our Rule" across all sentiment categories. It demonstrates a higher ability to capture the true positive instances for Positive, Negative, and Neutral sentiments. The F1-Score, which considers both Precision and Recall, shows a similar trend. "Sentiment Bert" achieves higher F1-Scores for all sentiment categories, indicating a better balance between Precision and Recall.

Overall, the results suggest that "Sentiment Bert" performs better than "Our Rule" in terms of Precision, Recall, and F1-Score for sentiment analysis. It demonstrates higher accuracy in sentiment classification and better ability to capture both positive and negative instances accurately. It's important to note that the performance of sentiment analysis models can vary depending on the dataset, training approach, and other factors. The comparison presented here is based on the specific evaluation results provided in the table and may not reflect the performance in other scenarios. Further analysis and evaluation on larger datasets and different domains would provide a more comprehensive understanding of the models' performance and generalizability.

4.3 Ablation Study on different rules

In analyzing the dataset for a rule-based aspect sentiment analysis system, several trends can be observed as the number of rules increases.

1. **F1-score**:

   - For the *negative* class, there is a noticeable upward trend in the F1-score as the number of rules increases. This indicates that adding more rules progressively improves the harmonic mean between precision and recall for the negative class. However, it is important to note that the rate of improvement diminishes slightly after the 7th rule.

Table 4: Performance Metrics vs. Number of Rules

Number of Rules	Negative			Neutral			Positive		
	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
1	0.128	0.145	0.114	0.808	0.933	0.712	0.217	0.260	0.186
2	0.352	0.425	0.301	0.722	0.833	0.637	0.433	0.518	0.373
3	0.373	0.445	0.321	0.702	0.813	0.617	0.454	0.538	0.393
4	0.404	0.465	0.358	0.682	0.787	0.602	0.486	0.558	0.430
5	0.425	0.485	0.378	0.658	0.757	0.582	0.514	0.598	0.450
6	0.445	0.505	0.398	0.632	0.725	0.561	0.532	0.623	0.464
7	0.463	0.522	0.416	0.613	0.705	0.542	0.546	0.651	0.470
8	0.463	0.522	0.416	0.613	0.705	0.542	0.546	0.651	0.470
9	0.503	0.568	0.451	0.595	0.685	0.526	0.579	0.691	0.499
10	0.506	0.571	0.454	0.595	0.685	0.526	0.580	0.692	0.500
11	0.544	0.614	0.488	0.561	0.647	0.496	0.617	0.736	0.530

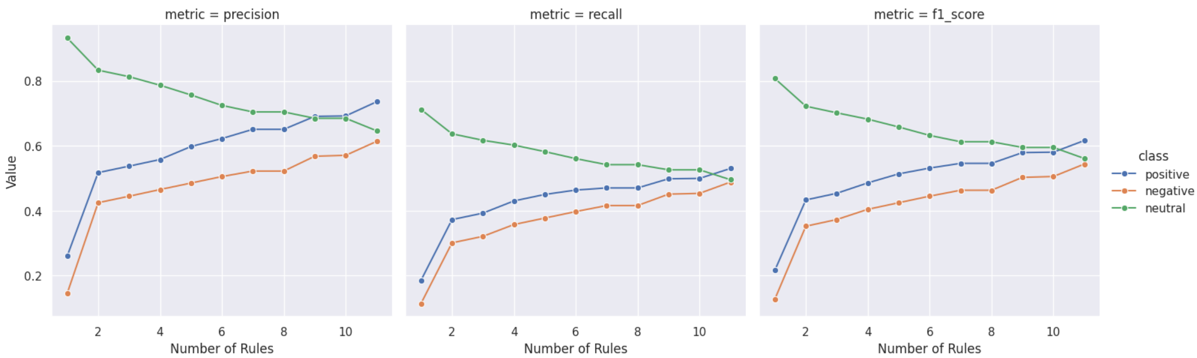


Figure 4: Ablation Study on different rules

- For the *neutral* class, there is a downward trend in the F1-score as more rules are added. Initially, with one rule, the F1-score is considerably high but starts to decrease consistently as the system complexity increases. This could indicate that adding more rules might be introducing some noise or overfitting for the neutral class.
- The *positive* class also shows an upward trend similar to the negative class, albeit at a slower rate.

2. **Precision**:

- The precision of the *negative* class shows an increasing trend as the number of rules increases. However, the rate of increase seems to be slowing down, indicating a possible saturation point.
- The *neutral* class, on the other hand, shows a decreasing trend in precision as more rules are added. This could mean that while trying to classify more examples correctly, the system is also producing more false positives.
- Similar to the negative class, the *positive* class exhibits an increasing trend in precision. This indicates that additional rules are helping to correctly identify positive instances with fewer false positives.

3. **Recall**:

- The recall for the *negative* class exhibits an increasing trend, though not as steep as the precision. This suggests that the system is becoming more sensitive to the negative class but not at the same rate as the positive class.

- For the *neutral* class, there is a consistent downward trend in recall as the number of rules increases. This indicates a reduction in the system's ability to correctly identify all the relevant neutral instances.
- In contrast, the *positive* class shows an increasing trend in recall, similar to precision, indicating that the system is identifying more true positives as more rules are added.

In conclusion, incrementally adding rules appears to enhance the system's ability to correctly classify negative and positive sentiments, as evidenced by increases in F1-score, precision, and recall. However, it seems to have an adverse effect on the neutral class. The trends indicate that there is a trade-off between complexity and performance, and that careful consideration should be given to the number of rules incorporated into the system to avoid overfitting or diminishing returns. Additionally, further investigation may be required to understand why the neutral class is adversely affected, and how this can be mitigated.

4.4 Failure Case Study

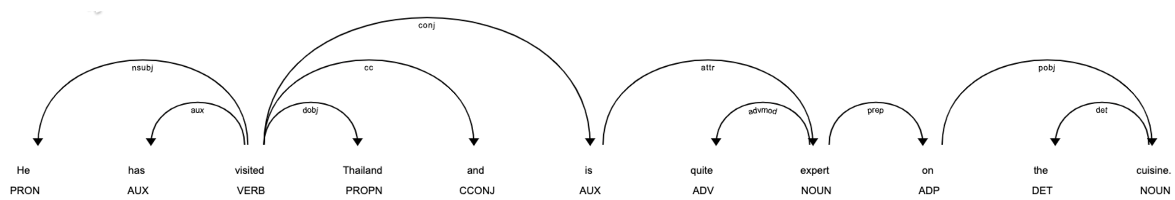


Figure 5: Failure case

In the analysis of the input text, "He has visited Thailand and is quite expert on the cuisine," there are two aspect terms that warrant attention, one of which is "cuisine" which should be associated with a positive sentiment.

Commencing the analysis with the term "cuisine", it is observed that this term is characterized by the grammatical attribute of being a NOUN. However, according to Rule 10 which concerns sentiment self-assessment, this term does not inherently possess any sentiment value. An attempt to identify sentiment by examining its syntactic relationships is made by navigating to its head node, which is the word on. The traversal of all child nodes of "on" does not yield any sentiment value either. Further navigation to the head node of "on", which is "expert", also fails to yield sentiment information as the node "expert" does not inherently carry sentiment meaning, and its children do not convey positive sentiment.

Upon human evaluation, it is discerned that the positive sentiment associated with "cuisine" can be attributed to the combined effect of the phrase "quite expert". This implies that a more composite analysis taking into consideration the interdependence of words is required.

These observations suggest that a notable portion of the failure cases in sentiment parsing may be attributed to the inadequacies of the sentiment parsing system. Even though attempts were made to leverage a pre-trained library in lieu of a direct polarity dictionary, there was no significant improvement in performance. This indicates that the precise identification of sentiment and polarity remains a challenging research question within the realm of natural language processing, especially for cases that involve the syntactic and semantic relationships between words.

5 Conclusion

In this study, we embarked on the endeavor to construct an aspect-based sentiment analysis system employing dependency parsing and syntactic analysis. To achieve this objective, we meticulously developed a feature-rich, highly optimized framework capable of processing multiple rules for any given sentence and executing aspect sentiment analysis corresponding to specific aspect terms. A pivotal attribute of our system is its extensibility, which is achieved through the judicious utilization of a chain mechanism.

However, upon evaluation, it was observed that the performance of our system falls short when juxtaposed with sentiment analysis systems that are underpinned by deep learning, such as those employing BERT (Bidirectional Encoder Representations from Transformers). One of the plausible reasons for this shortfall is the relatively inferior accuracy of our classifier.

Upon reviewing relevant literature and assessing contemporary methodologies, we posit that there are two potentially promising avenues for enhancing the performance of aspect-based sentiment analysis systems:

1. **Deep Learning Parser & Classifier** [2]: The first approach involves harnessing deep learning-based methodologies in lieu of dependency parsing and traditional sentiment analysis. This could entail pre-training a sizable neural network on a diverse dataset. Through this process, the model would be able to learn feature representations and sentiment classifications more nuancedly. The capacity of deep learning to capture complex, high-dimensional relationships within the data may contribute to more robust and accurate sentiment analysis.

2. **Rule-based System through Reinforcement Learning** [11]: The second approach involves the development of a rule-based aspect sentiment analysis system, augmented with reinforcement learning. In this paradigm, a policy and reward mechanism could be established to optimize the process of rule construction. By learning from feedback and iteratively refining the rules, the system can adapt to the complexities of natural language, potentially improving the accuracy and efficacy of sentiment extraction.

In summary, as the landscape of natural language processing continues to evolve, exploring and integrating innovative methodologies, such as deep learning and reinforcement learning, is imperative to address the intricate challenges associated with aspect-based sentiment analysis. Such advancements could pave the way for more sophisticated and reliable systems that capture the subtleties and nuances inherent in human language.

References

[1] Himanshu Batra, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Bert-based sentiment analysis: A software engineering perspective. In *Database and Expert Systems Applications: 32nd International Conference, DEXA 2021, Virtual Event, September 27–30, 2021, Proceedings, Part I 32*, pages 138–148. Springer, 2021.

[2] Hai Ha Do, Penatiyana WC Prasad, Angelika Maag, and Abeer Alsadoon. Deep learning for aspect-based sentiment analysis: a comparative review. *Expert systems with applications*, 118:272–299, 2019.

[3] Giuseppe DAniello, Matteo Gaeta, and Ilaria La Rocca. Knowmis-absa: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis. *Artificial Intelligence Review*, 55(7):5543–5574, 2022.

[4] Mohammad Erfan Mowlaei, Mohammad Saniee Abadeh, and Hamidreza Keshavarz. Aspect-based sentiment analysis using adaptive aspect-based lexicons. *Expert Systems with Applications*, 148:113234, 2020.

[5] Baris Ozyurt and M Ali Akcayol. A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: Ss-lda. *Expert Systems with Applications*, 168:114231, 2021.

[6] Ioannis Pavlopoulos. Aspect based sentiment analysis. *Athens University of Economics and Business*, 2014.

[7] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8600–8607, 2020.

[8] Minh Hieu Phan and Philip O Ogunbona. Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3211–3220, 2020.

[9] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics, 2016.

[10] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*, 2016.

[11] Burrhus Frederic Skinner. *Contingencies of reinforcement: A theoretical analysis*, volume 3. BF Skinner Foundation, 2014.

[12] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. Relational graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.12362*, 2020.

[13] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv preprint arXiv:1603.06679*, 2016.

[14] Nurulhuda Zainuddin, Ali Selamat, and Roliana Ibrahim. Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence*, 48:1218–1232, 2018.

[15] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2022.