

# Are You Sure You’re Positive? Consolidating Chain-of-Thought Agents with Uncertainty Quantification for Aspect-Category Sentiment Analysis

Filippos Ventirozos<sup>1,2</sup>, Peter Appleby<sup>2</sup>, Matthew Shardlow<sup>1</sup>

<sup>1</sup>Manchester Metropolitan University,

<sup>2</sup>Autotrader Research Group, Autotrader UK

Correspondence: [f.ventirozos@mmu.ac.uk](mailto:f.ventirozos@mmu.ac.uk)

## Abstract

Aspect-category sentiment analysis provides granular insights by identifying specific themes within product reviews that are associated with particular opinions. Supervised learning approaches dominate the field. However, data is scarce and expensive to annotate for new domains. We argue that leveraging large language models in a zero-shot setting is beneficial where the time and resources required for dataset annotation are limited. Furthermore, annotation bias may lead to strong results using supervised methods but transfer poorly to new domains in contexts that lack annotations and demand reproducibility. In our work, we propose novel techniques that combine multiple chain-of-thought agents by leveraging large language models’ token-level uncertainty scores. We experiment with the 3B and 70B+ parameter size variants of Llama and Qwen models, demonstrating how these approaches can fulfil practical needs and opening a discussion on how to gauge accuracy in label-scarce conditions.

## 1 Introduction

Tracking customer satisfaction is critical for organisations aiming to improve their products and services. However, traditional supervised approaches require bespoke datasets, necessitating time and human labour (Pustejovsky and Stubbs, 2013). The costs of annotation can be a significant challenge for many organisations, prohibiting access to adapted state-of-the-art solutions, and even to evaluation. Hence, in this paper, we explore and compare two popular large language models (LLMs) across various low-training scenarios, specifically zero-shot, for analysing customer satisfaction.

Sentiment analysis approaches are commonly used to classify feedback as positive, neutral, or negative for customer satisfaction, but this method often lacks granularity. A single piece of feedback

can express multiple sentiments targeting different product or service aspects, leading to ambiguous classifications and limiting actionable insights. To address this, aspect-based sentiment analysis (ABSA) methodologies (Zhang et al., 2023) have been developed to more accurately capture the nuanced sentiments present in customer feedback.

In our study, we investigated the ABSA task of aspect category sentiment analysis (ACSA), a type of ABSA. For a given piece of text, ACSA considers the aspect categories and sentiment polarities, as shown in Figure 1. The categories are a set of classes which are predefined by domain experts or stakeholders. Characteristically, in the figure, the example demonstrates two tuples from the same text. Firstly, the review found the pepperoni pizza, a type of #Food (Category), to be delicious, representing a positive sentiment (Polarity). Secondly, the review found the service to be terrible, representing negative polarity.

ACSA Tuple Example		
<b>Input:</b> <i>The pepperoni pizza was delicious but the service was terrible though.</i>		
<b>Output:</b>	<b>Category</b>	<b>Polarity</b>
	#Food	positive
	#Service	negative

Figure 1: An ACSA example where a review results in two category-sentiment polarity pairs.

Supervised learning approaches hold the state of the art for ACSA tasks (Cai et al., 2020; Ping et al., 2024; Xu et al., 2025). In contrast, we explore the zero-shot setting, which eliminates the need for labelled instances and allows us to experiment with LLM agents on novel datasets. This approach is particularly compelling in real-world applications,

as organisations and firms often lack the resources or time required to annotate large datasets.

In this paper, we propose a novel approach that integrates multiple LLM agents, each utilising a distinct chain-of-thought (CoT) reasoning process. Prior research (Fei et al., 2023; Wang and Luo, 2023) has introduced a sequential CoT approach, where aspects are first identified, followed by opinions and then polarities. In our experiments, we adapt this framework to the ACSA task and critically evaluate the assumptions underlying CoT by exploring alternative reasoning sequences. Additionally, we introduce mechanisms to combine the outputs of each CoT agent through aggregation techniques that make use of the LLMs’ token confidence scores, also referred to as token-level uncertainty.

The primary contributions of this study are enumerated below:

1. We experimented with how the ordering of CoT prompt elements affects LLMs’ performance across datasets.
2. We compared multiple aggregation methods for effectively combining outputs from multiple CoT agents.
3. We benchmark our approach against a previously published zero-shot method, reproduced and evaluated on new datasets.

## 2 Related Work

ABSA has emerged as a particularly interesting research topic due to its increasing popularity and widespread applicability across various domains (Rink et al., 2024; Namee et al., 2023; Kit and Joseph, 2023; Arianto and Budi, 2020; Chu et al., 2022; Yu et al., 2023 inter alia).

In this paper, we address the ACSA task, a type of ABSA. The ACSA task is especially relevant for industry applications, as it involves associating predefined category terms with corresponding sentiment polarities, which can be readily quantified and analysed (more on Section 3.2).

Although a few studies have considered the ACSA task from a supervised learning perspective (Cai et al., 2020; Ping et al., 2024; Xu et al., 2025), it would be beneficial to address it in an unsupervised manner because annotation requires considerable time and human labour, and may also obfuscate reproducibility, as discussed later in this paper.

Zero-shot and few-shot methodologies have been employed for the broader ABSA task. For instance, Hellwig et al. (2025) use LLMs with a few-shot in-context learning (ICL) approach, while Wang and Luo (2023) evaluate different prompting techniques for sentiment analysis using LLMs. Fei et al. (2023) investigated a multi-hop CoT approach, and Bai et al. (2024) introduced ChatABSA, an ICL few-shot prompt framework for ABSA that also extends to the ACSA task we address.

In our paper, we aim to push the boundaries of zero-shot learning by examining the use of multiple LLM agents provided with no prior knowledge. More agents can yield better results; indeed, Li et al. (2024) discuss how employing additional agents and subsequently performing majority voting can improve outcomes.

To the best of our knowledge, there is no prior work on multi-agent collaboration for ACSA. We therefore experiment with this approach, inspired by previous ABSA research on CoT (Fei et al., 2023; Wang and Luo, 2023), and we investigate different aggregation techniques that leverage token-level uncertainty in LLMs (Shorinwa et al., 2024).

## 3 Methodology

### 3.1 Problem Statement

Our problem statement aligns with prior work on ACSA. Given a text (whether a single sentence from a review or an entire review) our goal is to extract one or more pairs, each consisting of a category and its corresponding polarity, as depicted in Figure 1. While the order of the pairs may vary due to the generative nature of the models used, the internal structure of each pair must remain consistent: the category always precedes the sentiment polarity. This ensures clarity and uniformity in the extracted results. Such as:

$$Q = \{(c_i, p_i)\}_{i=1}^n \quad (1)$$

where  $Q$  represents the set of pairs,  $c_i$  is the category, and  $p_i$  is the polarity for the  $i$ -th pair, and  $n$  is the number of pairs extracted from the text. In the following subsections, we describe the different methods utilised in our experiments.

### 3.2 CoT Agents

Previous research on related ABSA tasks (Fei et al., 2023; Wang and Luo, 2023) have typically followed an intuitive sequence: first extracting aspects,

and then identifying opinions and their corresponding polarities. However, in our experiments, we challenge this conventional approach by exploring alternative reasoning pathways. Specifically, we design and evaluate multiple LLM agents, each guided by a distinct prompt that dictates a unique CoT process for extracting category-polarity pairs.

To better understand the ACSA task, let us break it down step by step. In a review, aspect terms are the words or phrases that refer to specific features of a product or service, such as “pepperoni pizza” in our example in Figure 1. While these aspect terms are helpful, they are not always ideal for quantitative analysis. For instance, rather than analysing sentiment for individual dishes (e.g., pepperoni pizza, pasta) that may vary across restaurants, it is more meaningful to map these terms to broader categories, such as “Food”, to measure overall sentiment on food for each restaurant. Similarly, opinions like “delicious” provide qualitative insights but are not easily quantifiable. Mapping these opinions to sentiment polarities—positive, neutral, or negative—enables more actionable and comparative analytics.

In our CoT experiments, we prompted the LLMs to detect one element at a time, ultimately generating category-polarity pairs. We systematically permuted the order of the three elements: aspect terms, aspect categories, and opinions. Sentiment polarity was always derived from the opinion generated at the end of the prompt. For example, one agent might follow the sequence *aspects* → *categories* → *opinions*, while another might use *opinions* → *aspects* → *categories*. For simplicity, we refer to the entities in the CoT sequences (i.e., aspects, categories, and opinions) as “elements”. This systematic variation of element ordering enables us to investigate how reasoning sequences affect the accuracy of sentiment analysis prediction.

### 3.2.1 Prompt Crafting

Our next step involved designing the necessary prompts to chain the reasoning elements effectively. Previous research has explored multi-hop CoT approaches [Fei et al. \(2023\)](#); [Wang and Luo \(2023\)](#), where the process involves using a prompt to request each element from the LLM sequentially with a new call. However, in our preliminary experiments (see Appendix H), we found that this multi-hop approach did not yield superior results compared to consolidating all instructions into a

single prompt providing enumerated instructions. Consequently, we adopted an enumerated CoT reasoning process within a single prompt. Examples of our prompts are illustrated in Appendix E, and our multi-hop CoT prompts adjusted for our experiments are in Appendix F.

The enumerated instructions inside the prompt were manually crafted in an imperative language, inspired by previous research in the space ([Fei et al., 2023](#); [Wang and Luo, 2023](#); [Bai et al., 2024](#)). To ensure that our final prompts are reasonably robust, we evaluated them on the inter-prompt stability score ([Barrie et al., 2025](#)), using the 3B Llama model as a reference. We noticed our prompt maintained a constant inter-prompt stability score across different ranges of temperatures (0.1, 0.5, 1.0, 2.0, 5.0) for paraphrasing the prompt with the encoder-decoder PEGASUS model ([Zhang et al., 2020](#)), where stability across temperatures for paraphrasing is optimal, whereas if it deviates, is not ([Barrie et al., 2025](#)).

In the last part of the prompt, we instructed the LLM to format the output as Python code, specifically as a list of pairs. Each pair consists of two elements: the category and the sentiment polarity. Additionally, we incorporated a system instruction, following the approach of prior research ([Wang and Luo, 2023](#)), which served as a role-playing mechanism. This instruction constrained the LLM to generate the most probable answers only, minimising verbosity.

### 3.2.2 Post-Processing

From the LLM’s text-generated response, we would parse the output string into Python code, neglecting any text generated outside the list. Then, we would use the native’s Python `diff` lib library<sup>1</sup> to map the category from the generated text to the list of potential categories. This was done to avoid discrepancies and spelling mistakes in generation when classifying the category-polarity pairs.

## 3.3 CoT Aggregation Techniques

After we obtained each agent’s results, we explored various techniques and algorithms for aggregating the resulting pairs of each agent. For most of our algorithms, we would rely on the token-level uncertainty produced by the LLMs.

Specifically, during generation, the LLMs assign each token a logarithmic probability based

<sup>1</sup>The `diffib` library works by finding the longest continuous characters between strings.

on the conditional distribution given all preceding tokens. For convenience, we convert these logarithmic values into probabilities. Subsequently, we extract the category and sentiment polarity words for each pair, omitting any special characters (i.e. the Python brackets), and compute the average of the probabilities of these words to obtain the final probability score for each pair, also referred to as the pair’s confidence score. Below we list the algorithms, where a list refers to a list of pairs generated by the language models:

1. **Highest probability list** Given that each agent has generated a list of pairs, we would acquire the top list in confidence score from the pool of agents, based on the averaged pair probabilities for a given list.
2. **Most common list** We acquire the most common list of pairs between agents. If there is no majority, then pick the agent(s) pairs with the highest confidence score.
3. **Highest probability pairs** We get the top  $n$  most probable pairs across all lists (i.e. across all agents). We also evaluated different techniques for setting the  $n$  for each prediction. More on that can be found in the Appendix A.
4. **Clustered pairs** After obtaining the number of pairs  $n$ , identically to the above algorithm, we cluster all the unique pairs from all the agents based on their category word embeddings into  $n$  clusters using the k-means algorithm from the Scikit-learn library (Pedregosa et al., 2011). These embeddings are generated by the RoBERTa encoder model (Liu et al., 2019), fine-tuned in a semantic textual similarity sentence setting (Reimers and Gurevych, 2019). Then, we select the pair with the highest probability score from each cluster. We chose a sentence embedding model as opposed to a word embedding one since one category can be multiple tokens, for instance, the category “performance#sizing/fit” from the Shoes dataset.
5. **Most confident agent** We select the agent whose predictions yield the highest sum of list confidence across the dataset, where the list confidence is again the average pair probabilities in that list.

In Figure 2 we show the workflow from the various CoT agents to funnelling through an aggregation technique to get the end list of pairs.

## 4 Experiments

### 4.1 Datasets

For our experiments, we included four datasets. We opted for the Laptop16 (Pontiki et al., 2016), Restaurant16 (Pontiki et al., 2016), MAMS (Jiang et al., 2019) and Shoes (Peper et al., 2024). All of these datasets were developed for the ACSA task, apart from the Shoes one, which was developed with an aspect-sentiment quad prediction task (Aspect, Category, Opinion, Sentiment) in mind, which we repurposed for ACSA (using only the Category and Sentiment labels).

Large-scale LLMs are known to be trained on existing code repositories and the open internet, which may include pre-existing datasets (Samuel et al., 2025). We purposely included the recent Shoes dataset since the LLM models do not exhibit any knowledge of this dataset, making it an ideal case to analyse performance on the totally “unseen” dataset. Moreover, the Shoes dataset is the only one which considers whole reviews as input, whereas the other datasets have segmented the reviews into sentences. In Appendix B, Table 2, we provide the number of instances for each data fold and the number of categories.

### 4.2 LLM Models

For our experiments, we used the 72B and 3B Qwen model (Qwen et al., 2025) and Llama 3B and 72B model (Llama et al., 2024). Since we want to compare the probabilities between tokens, we opted for the greedy search decoding strategy. The detailed versions of our models are listed in Listing 1.

- Qwen/Qwen2.5-72B-Instruct
- Qwen/Qwen2.5-3B-Instruct
- meta-llama/Llama-3.3-70B-Instruct
- meta-llama/Llama-3.2-3B-Instruct

Listing 1: LM versions used in our experiments.

### 4.3 Evaluation

The ACSA task focuses solely on extracting the category and polarity tuples. This task is best characterised as a multi-label classification problem, as the categories are predefined per domain and the polarity values range across positive, neutral, and negative. We adhere to the same micro-F1 metric as



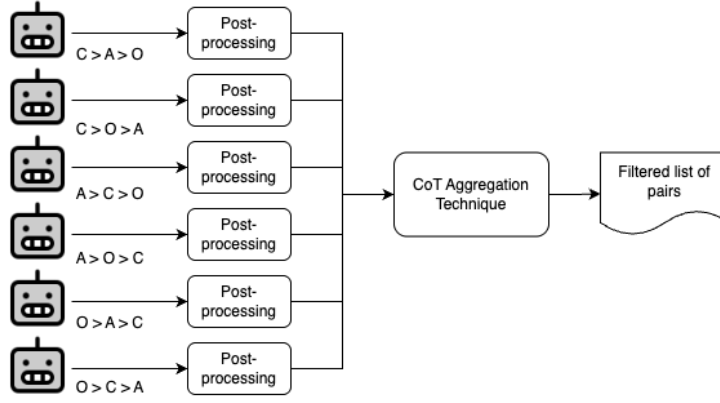


Figure 2: On the left side, we present the various CoT agents derived from the same LLM, each employing a different CoT extraction order. The letters C, A, and O denote Category, Aspect, and Opinion respectively, while the chevrons between them indicate their extraction sequence. The outputs from these agents are subsequently post-processed and passed through an aggregation technique to obtain the final set of ACSA pairs.

those used in the previously mentioned evaluations, as disclosed in Cai et al. (2020).

#### 4.4 Baseline

For benchmarking, we utilise the ChatABSA method (Bai et al., 2024). The authors of ChatABSA have designed a prompt specifically for the ACSA task, allowing the inclusion of a certain number of ICL samples as few-shot demonstrations. In our experiments, we adopt the zero-sample setting, as our goal is to perform comparisons in a zero-shot scenario.

## 5 Results & Discussion

**Element Order Debunking** We tested three element permutations for each LLM model (i.e., aspect, category and opinion). Our experiments on the four datasets demonstrated that the order of the three elements between the two larger LLM models was fairly consistent, but it differed for each dataset. For instance, order 1) Opinion 2) Category 3) Aspect produced higher F1 scores for the Laptop16 dataset, whereas order 1) Category 2) Aspect 3) Opinion was best for the MAMS dataset. For the other two datasets, although they differed, the position of the Opinion and the Categories always followed the same pattern, indicating a CoT trend per dataset.

When comparing the 3B LLMs, we notice the same pattern in that the CoT element order stays fairly consistent across each dataset. Moreover, for all the LLMs, we did not notice any CoT agent that consistently underperformed; similarly, it varied across datasets but not across the models. Thus,

we draw two conclusions from our experiments. Firstly, as opposed to previous research (Fei et al., 2023; Wang and Luo, 2023), starting from the aspects in the CoT prompts does not always provide optimal results. Secondly, the order is rather data-dependent and stays consistent across our four models. In the case of our multi-hop CoT preliminary experiments, the best-performing agents always had the Opinion preceding the Category and Aspect, see Table 3. For more detailed results across the agents per model and dataset, one can look in the Appendix C.

**Joined CoT Agent** We combined all pairs generated by the six CoT agents and refer to this combination as the *joined CoT agent*. The results indicate that, as expected, precision decreased, but recall was higher than that of any individual CoT agent. Notably, for our models with 70B+ parameters, the average recall improvement was approximately 20% across all datasets, except for the Restaurant16 dataset, since in that particular dataset, prediction scores were already relatively high. The 3B models did not substantially lag behind in recall scores with their larger counterparts. These findings illustrate the potential benefit of leveraging multiple agents from different CoT chains to enhance performance in ABSA tasks.

To better understand the performance of each LLM, we looked at all the aggregated pairs. It was evident that the smaller language models produced many more pairs, many of which were irrelevant. Especially the 3B Llama produced in one dataset 6x the number of pairs than its bigger 70B sibling, as seen in Figure 3. Moreover, we counted the num-

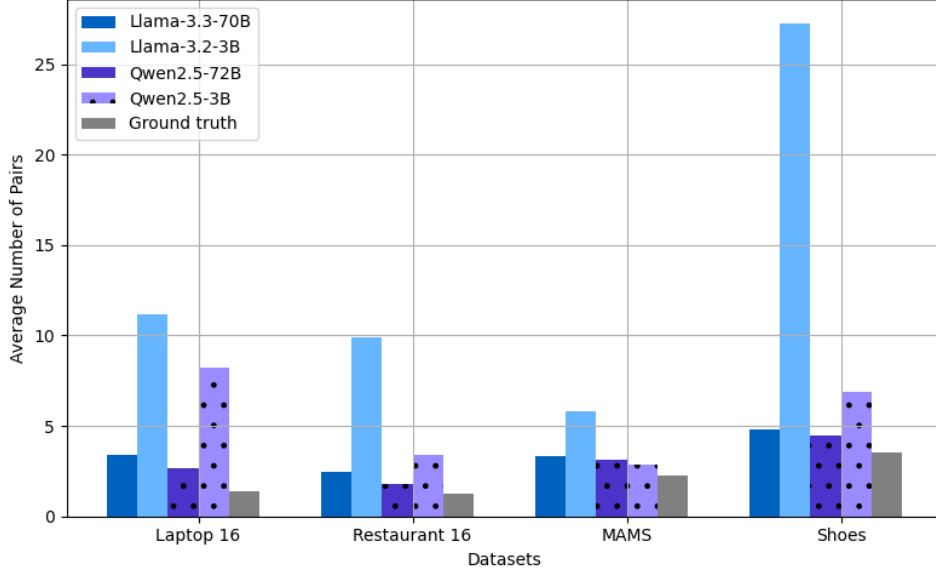


Figure 3: The average number of pairs generated for each model for each dataset. These refer to the *Joined CoT Agent*, the pairs are summed. In dark grey is the average number of the ground truth for each dataset.

ber of pairs where the category was the same, but there was more than one sentiment polarity with the same category. The 3B models, with Llama leading, demonstrated many more conflicting categories, whereas the larger ones were at a minimum, suggesting that the smaller LLMs struggled to delineate the polarities. One can view the number with conflicting pairs in Appendix D.

**CoT Aggregation Techniques** Subsequently, we compared the various aggregation techniques, as enumerated in Section 3.3. The average results are presented in Table 1. It can be observed from the table that the *highest probability list* provided the best overall results amongst all the aggregation techniques. These results suggest that the confidence provided by the LLMs can be useful.

Following this, the *most common tuples* among agents scored second, suggesting that relying on confidence scores is better than majority voting. Unfortunately, selecting pairs apart from the lists was not the most effective strategy: any strategy in this category—*highest probability pairs* and *clustered pairs* with any  $n$  technique—underperformed compared to the two strategies that left the lists intact, although they appeared to achieve higher scores in the 3B model space. We speculate that this is because LLMs inherently can approximate the ground truth list of pairs, but they may deviate due to different interpretations that stray from the ground truth.

**LLMs Confidence Interpretation** The *highest confidence list* consistently emerged as the top aggregate technique when using the 72B Qwen; only, in Llama’s 70B case in the Restaurant-16 dataset, the *most common list* proved a better strategy. Conversely, while the *lowest confidence list* underperformed significantly with Qwen, it did not score as poorly with Llama. Moreover, when scoring the *most confident agent* amongst the datasets, Qwen’s agents two times matched the top performant agent, whereas in Llama’s case, it was once. These observations could suggest that the log probabilities provided by Qwen are more informative than those from Llama. The Qwen model is slightly larger, with 72 billion parameters compared to Llama’s 70 billion, but it is unclear if this is the sole reason for the observed differences.

One question that we posed to answer was whether the LLM’s top performant CoT agent was also the *most confident agent* (i.e. the agent with the sum of highest probability scores). Our results show that although the *most confident agent* would always be amongst the first, it was not always the top performer. In Llama’s case, the 70B model only in the MAMS dataset, the top performer was also the most confident, and in Qwen’s 72B, two out of four datasets, the top performer was also the most confident. It certainly demonstrates notable consistency, given that there are six different CoT agents that can achieve the highest score. Then, for the 3B LLMs, we got more fuzzy results, with

LLM Configuration	Qwen2.5-72B-Instruct	Qwen2.5-3B-Instruct	Llama-3.3-70B-Instruct	Llama-3.2-3B-Instruct
ChatABSA fs:0	48.5%	27.3%	50.4%	10.5%
Highest prob. list	<sup>2</sup> 55.5%	<sup>1</sup> 36.7%	<sup>2</sup> 54.3%	23.4%
Most conf. agent	54.2%	<sup>3</sup> 35.0%	53.2%	28.5%
Most common. list	<sup>3</sup> 54.4%	34.0%	<sup>3</sup> 53.9%	<sup>1</sup> 29.7%
Highest prob. pairs alpha:1	52.9%	33.0%	50.5%	26.3%
Highest prob. pairs alpha:0.9	52.9%	32.8%	50.5%	<sup>3</sup> 26.8%
Highest prob. pairs alpha:mean	51.9%	31.0%	49.7%	25.6%
Highest prob. pairs alpha:max	51.4%	<sup>2</sup> 34.4%	50.0%	<sup>2</sup> 27.2%
Top performant agent	<sup>1</sup> 56.6%	36.0%	<sup>1</sup> 54.6%	26.8%

Table 1: The table demonstrates the F1 scores averages in percentages for each model averaged on our four datasets. The first one, Chat-ABSA, in the zero-shot setting, is our baseline. Following are our top 7 aggregation techniques, and lastly is the CoT agent, which scored the highest on each dataset, and then averaged. The top three F1 scores for each model are coloured in podium finish and annotated with superscripts (<sup>1</sup>gold, <sup>2</sup>silver, and <sup>3</sup>bronze) for accessibility.

Llama having three out of four and Qwen none out of the four datasets.

To better understand the token-level confidence allocation of each LLM, we analysed the confidence scores of the two larger LLMs. We averaged the probabilities of each pair and then calculated the Spearman correlation (Spearman, 1904) between the scores averaged by each agent and the corresponding F1-score; we performed the same analysis using variance instead of averages. We observed a positive correlation between the averaged probabilities across agents and the F-scores, suggesting that confidence scores indeed reflect a perceived measure of accuracy (see Figure 4). Characteristically, Llama exhibited the highest correlation, achieving nearly 60% correlation on certain datasets. The scores on the MAMS dataset demonstrated a weaker correlation; this could be attributed to the more challenging examples, where sentences contain multiple aspects expressing conflicting emotions (Jiang et al., 2019).

Moreover, as shown in Figure 5, the variance of probabilities across agents exhibited precisely the same trend as the averaged scores, but inverted. This observation suggests that greater variance among agents is likely associated with a lower F-score.

**Few Shot CoT** We evaluated the performance of ChatABSA with 10 few-shot demonstrations against CoT combined with the same 10 few-shot examples. In Appendix G, we provide further details on how we adapted our CoT prompt to accommodate these few-shot examples.

As expected, when we added 10 examples to ChatABSA, the F-score percentage increased across all datasets. When averaged and compared

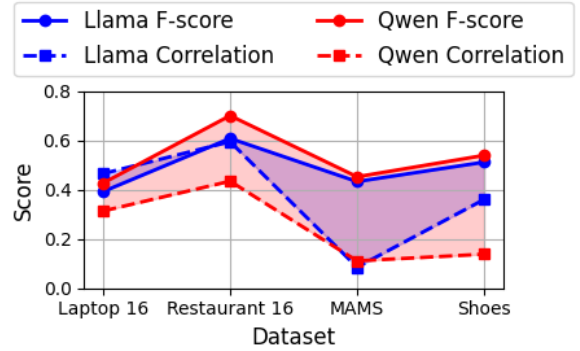


Figure 4: Comparison of F-score and Spearman correlation on averaged confidence and F-Score for two LLM models across datasets. Shaded areas represent the difference between the two metrics.

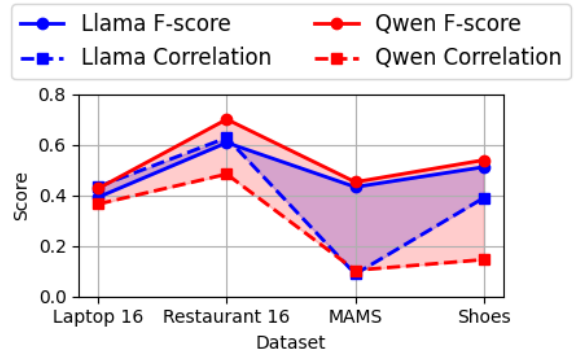


Figure 5: Comparison of F-score and Spearman correlation on confidence variance and F-Score for two LLM models across datasets. Shaded areas represent the difference between the two metrics. The values of the correlation here are swapped with positives. Hence, a correlation in the positive of 0.4 would mean -0.4.

to Table 1, ChatABSA exhibited notable improvements, particularly for the Llama models, with the 3B version increasing by 38% and the 70B version by 15%. For the Qwen models, we observed increases of 22% for the 3B model and 11% for the 72B model. However, incorporating examples into the CoT resulted in only modest changes: a 4% increase for Llama 70B, a 3% decrease for Llama 3B, a 2% increase for Qwen 72B, and an 8% increase for Qwen 3B. We observed that the Llama models benefited more substantially from the few-shot demonstrations alone, while the combination of elaborate CoT prompts and examples appeared to “confuse” the 3B Llama model. Notably, the Llama 70B model became the only case in which ChatABSA surpassed CoT when averaged across datasets, given the same ten examples as demonstrations.

We hypothesise that the advantage of clear few-shot demonstrations without CoT is that it allows “the LLMs to focus” directly on the provided examples and any biases they may introduce without the additional cognitive load imposed by the CoT process, which may divert the model’s attention from effectively processing these examples.

**CoT Prompt Stability Validation** We conducted a validation experiment to see the effect of the prompt, comparing the engineered version of the few-shot CoT but without examples (see Appendix G) to the previously mentioned CoT prompt version (see Appendix E). We ran all models across all datasets with the same system instruction. We observed that, even after changing the prompt, the F1 scores of the 70B+ models remained consistent, differing only from the third decimal point onward. In contrast, for the 3B models, differences appeared from the second decimal point.

**Practical Insights** From Table 1, we can see that the *Top performant agent* for each dataset dominates in the higher parameter size LLM models. However, without labelled data, evaluating which CoT order sequence works best for a dataset would be difficult. Following, the ***Highest probability list demonstrated favourable results***, indicating that this would be a good choice if no labelled data is available.

We would recommend using various CoT agents and checking the consistency on an unexplored dataset. In Paragraph 5, we demonstrated that **high confidence variance between the agents indicates a more challenging task**. One could

leverage this insight to identify challenging reviews within their dataset and examine why the agents’ confidence scores deviate among them.

Another valuable metric to include in one’s evaluation repertoire is the extent to which different CoT agents produce **conflicting categories** (e.g., a category that one agent detects as positive while another agent detects it as negative). As observed, smaller models exhibit this issue more frequently, whereas larger LLMs tend to generate fewer conflicts and consequently achieve higher performance.

Relying on CoT reasoning without input examples offers several advantages over few-shot ICL and fully-supervised approaches. One key benefit is the reduction of biases that can arise from selecting examples. Popular publicly available datasets annotated by multiple crowd-sourced annotators often suffer from annotation bias (Geva et al., 2019; Gururangan et al., 2018; Paullada et al., 2021). Moreover, within large organisations, if an individual sets the initial examples to address specific cases, these examples may not remain relevant over time. In a dynamic business environment, products, their aspects, and even the workforce evolve, leading to changes in the context and nature of customer feedback. If the initial examples are not regularly updated, they can introduce biases or lead to outdated groupings, ultimately compromising the accuracy and relevance of the analysis.

By contrast, the **zero-shot approaches do not rely on predefined examples, allowing the model to adapt flexibly to changing contexts**. This approach promotes a more generalised and robust framework for sentiment analysis, reducing the risk of biases and ensuring that the model remains aligned with the current business environment. This adaptability is particularly valuable in scenarios where frequent updates to training data or examples are impractical or resource-intensive.

Another important consideration when applying a zero-shot LLM approach to the ACSA task is **avoiding overlapping categories in their annotation schema**. It became evident that certain labels tend to be confused; for instance, *contextofuse#purchase\_context* with *general* in the Shoes dataset, *miscellaneous* with *food* in MAMS, *laptop#miscellaneous* with *laptop#general* in Laptop16, and *food#style options* with *food#quality* in Restaurant16, among numerous other examples. These overlaps suggest that it is challenging to delineate whether a detected category should fall



under an umbrella term (e.g., *general*) or among the more specific labels without access to annotation guidelines or a view of the possible annotators' biases. We recommend that distinct labelling with non-overlapping categories be required when implementing a zero-shot approach.

The advantage of few-shot or fully supervised learning is that it can capture annotators' biases and label the data accordingly. However, we raise the question of how practical this approach is for industry stakeholders who lack the resources to annotate data extensively and require rapid and reproducible results. Therefore, we advocate for the adoption of zero-shot implementations.

## 6 Future Direction

The results demonstrated that the joined CoT agents achieved quite high recall, outperforming any individual CoT agent. This demonstrates that there is potential for improvement in the use of agents. One possible direction for future research could be to iteratively guide the inference process until a satisfactory summed log probability is reached. Moreover, one can try combining other CoT agents to evaluate possible improvements. Or utilising other aggregating techniques to filter the pairs.

Research has raised concerns regarding the reliability of token-level uncertainty, as token conditional distributions may sometimes be misleading. This issue arises particularly in cases where an initial token is incorrect, yet all subsequent tokens appear highly probable given that initial token (Shorinwa et al., 2024). Investigating more robust methods for estimating token-level confidence scores could improve the downstream ACSA performance or other tasks as well.

## 7 Conclusion

In our experiments, we evaluated two larger (70B+) LLMs, Qwen and Llama, along with their smaller 3B parameter-size counterparts. We observed that the optimal CoT order for the ACSA task is rather dataset-dependent. Additionally, we found that employing a multi-hop CoT approach does not yield better results compared to our enumerated CoT version. Furthermore, confidence scores derived from token-level uncertainty proved somewhat informative, appearing to be a more effective strategy than relying solely on majority voting among agents. Moreover, we discovered that using the variance in

confidence between agents can be a good indicator to view challenging examples. Intuitively, larger models tend to produce more reliable confidence estimates. Given the high recall achieved by our combined CoT aggregation method, we suggest there remains considerable room for improvement, potentially through exploring alternative aggregation techniques or developing more accurate token-level uncertainty estimation methods.

## 8 Limitations

One limitation is the primacy bias or order bias effect (Wang et al., 2023). LLMs' generation may get biased towards an earlier exposed option when asked, rather than a later one. This bias is particularly prevalent in category selection; when we ask the LLM which categories are mentioned in the text, it may have a tendency to select options presented at the beginning, ideally, we would need to change the order of the categories, but this would be computationally expensive to investigate.

## 9 Ethical Statement

LLMs, due to their training on extensive amounts of internet data, may inadvertently perpetuate and amplify biases present in the source material. This could result in biased or unfair outcomes in ABSA tasks, particularly in sensitive domains such as social media monitoring.

Our research contributes to the enhancement of sentiment analysis techniques, which can be beneficial in various fields such as marketing, customer service, and social sciences. By providing more accurate sentiment analysis, organisations and researchers can make better-informed decisions based on public and customer opinions.

However, there is a risk that enhanced ABSA techniques could be misused for unethical purposes, such as manipulating public opinion or spreading misinformation. We advocate for responsible use and adherence to ethical standards in deploying such technologies.

## References

- Dian Arianto and Indra Budi. 2020. [Aspect-based sentiment analysis on Indonesia's tourism destinations based on Google maps user code-mixed reviews \(study case: Borobudur and prambanan temples\)](#). In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 359–367, Hanoi, Vietnam. Association for Computational Linguistics.

- Yinhao Bai, Zhixin Han, Yuhua Zhao, Hang Gao, Zhuowei Zhang, Xunzhi Wang, and Mengting Hu. 2024. [Is compound aspect-based sentiment analysis addressed by LLMs?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7836–7861, Miami, Florida, USA. Association for Computational Linguistics.
- Christopher Barrie, Elli Palaologou, and Petter Törnberg. 2025. [Prompt stability scoring for text annotation with large language models](#). *Preprint*, arXiv:2407.02039.
- Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. 2020. [Aspect-category based sentiment analysis with hierarchical graph convolutional network](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 833–843, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- M. Chu, Y. Chen, L. Yang, and J. Wang. 2022. [Language interpretation in travel guidance platform: Text mining and sentiment analysis of tripadvisor reviews](#). *Frontiers in Psychology*, 13.
- Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. [Reasoning implicit sentiment with chain-of-thought prompting](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1171–1182, Toronto, Canada. Association for Computational Linguistics.
- Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. [Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Nils Constantin Hellwig, Jakob Fehle, Udo Kruschwitz, and Christian Wolff. 2025. [Do we still need human annotators? prompting large language models for aspect sentiment quad prediction](#). *ArXiv*, abs/2502.13044.
- Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. [A challenge dataset and effective models for aspect-based sentiment analysis](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6280–6285, Hong Kong, China. Association for Computational Linguistics.
- Brentton Wong Swee Kit and Minnu Helen Joseph. 2023. [Aspect-based sentiment analysis on movie reviews](#). In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, pages 237–243.
- Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. [More agents is all you need](#). *Preprint*, arXiv:2402.05120.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *Preprint*, arXiv:1907.11692.
- Llama, :, Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, and 77 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Khanista Namee, Jantima Polpinij, and Bancha Luaphol. 2023. [A hybrid approach for aspect-based sentiment analysis: A case study of hotel reviews](#). *Current Applied Science and Technology*, 23(2):Published: Aug 15, 2022.
- Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton, and Alex Hanna. 2021. [Data and its \(dis\)contents: A survey of dataset development and use in machine learning research](#). *Patterns*, 2(11):100336.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. [Scikit-learn: Machine learning in python](#). *J. Mach. Learn. Res.*, 12:2825–2830.
- Joseph J Peper, Wenzhao Qiu, Ryan Bruggeman, Yi Han, Estefania Ciliotta Chehade, and Lu Wang. 2024. [Shoes-ACOSI: A dataset for aspect-based sentiment analysis with implicit opinion extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15477–15490, Miami, Florida, USA. Association for Computational Linguistics.
- Zhichao Ping, Guoming Sang, Zhi Liu, and Yijia Zhang. 2024. [Aspect category sentiment analysis based on prompt-based learning with attention mechanism](#). *Neurocomputing*, 565.

- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. [SemEval-2016 task 5: Aspect based sentiment analysis](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.
- J. Pustejovsky and Amber Stubbs. 2013. *Natural language annotation for machine learning*. O’Reilly Media, Inc.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Lois Rink, Job Meijdam, and David Graus. 2024. [Aspect-based sentiment analysis for open-ended HR survey responses](#). In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 16–26, St. Julian’s, Malta. Association for Computational Linguistics.
- Vinay Samuel, Yue Zhou, and Henry Peng Zou. 2025. [Towards data contamination detection for modern large language models: Limitations, inconsistencies, and oracle challenges](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5058–5070, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. 2024. [A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions](#). *Preprint*, arXiv:2412.05563.
- Charles Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101.
- Yajing Wang and Zongwei Luo. 2023. [Enhance multi-domain sentiment analysis of review texts through prompting strategies](#). In *2023 International Conference on High Performance Big Data and Intelligent Systems (HDIS)*, pages 1–7.
- Yiwei Wang, Yujun Cai, Muhao Chen, Yuxuan Liang, and Bryan Hooi. 2023. [Primacy effect of ChatGPT](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 108–115, Singapore. Association for Computational Linguistics.
- Guixian Xu, Zhe Chen, and Zixin Zhang. 2025. [Aspect category sentiment analysis based on pre-trained bilstm and syntax-aware graph attention network](#). *Scientific Reports*, 15.
- Yang Yu, Duy-Tai Dinh, Ba-Hung Nguyen, Fangyu Yu, and Van-Nam Huynh. 2023. [Mining insights from esports game reviews with an aspect-based sentiment analysis framework](#). *IEEE Access*, 11:61161–61172.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*. JMLR.org.
- Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. [A survey on aspect-based sentiment analysis: Tasks, methods, and challenges](#). *IEEE Transactions on Knowledge and Data Engineering*, 35.

## A Number of Pair Estimation - Alpha

For the aggregation techniques *highest probability pairs* and *clustered pairs* described in Section 3.3, we developed several methods to determine the number of pairs to collect from the CoT agents for a given instance; we denote this variable as  $\alpha$ . These methods are enumerated as follows:

**Float Variable** Given a float variable, it would act as a variable of bias between the number of pairs produced by the CoT agents and the global average of the number of pairs produced in the whole dataset. In the below equation, the  $\mu$  represents the median of all the pairs on all the instances produced by the CoT agents, and the left part of the equation represents the average number of pairs for the current instance.

$$\text{round} \left( \alpha \cdot \frac{1}{N} \sum_{i=1}^N n_{\text{pairs},i} + (1 - \alpha) \cdot \mu \right)$$

In our study, we experimented with the values 0.9 and 1.

**String Variable** The other process involves first computing the average (list) probability for each agent and then associating this average with the number of pairs the agent produces. For instance, if an agent produces two pairs and has an average

probability of 0.9, we record the mapping (2, 0.9). Once we collect these (number of pairs: average log probability) mappings for all agents, we subsequently apply either the *mean* or *max* operation to the mapped probabilities. Finally, we select the number of pairs with the highest value. This illustrates how the *mean* and *max* variables operate when configuring the *alpha* score.

## B Dataset Statistics

In Table 2 we show the dataset statistics.

## C CoT Agent Performance

In Figures 6, 7, 8, 9 we show the various CoT F1 scores per dataset, as shown in the legend. Each figure is a different model. The letters A, C and O stand for Aspects, Category and Opinion, respectively and the arrows in between denote their order in the CoT prompt.

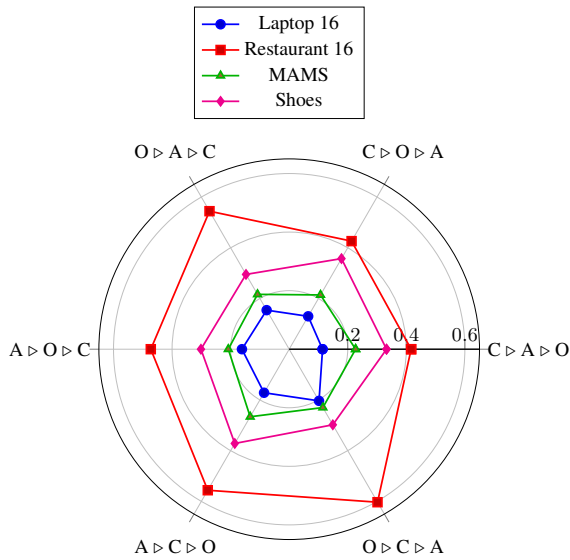


Figure 6: Radar plot comparing F-scores across different datasets and element orders for the Qwen 3B model.

## D Joined CoT Agents

Below, in Figure 10, we report the number of conflicting pairs for each dataset, after combining the pairs produced by the CoT agents, also referred to as the *Joined CoT Agent*.

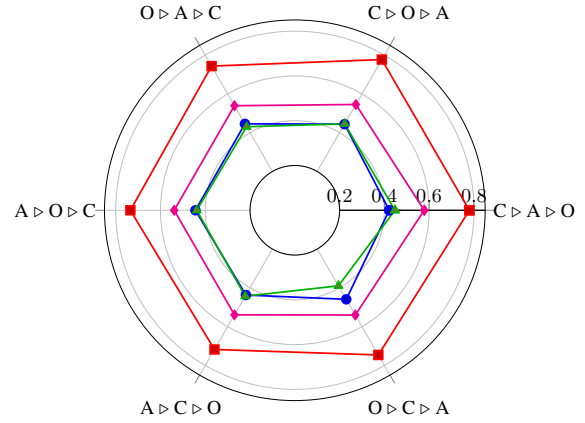


Figure 7: Radar plot comparing F-scores across different datasets and element orders for the Qwen 72B model.

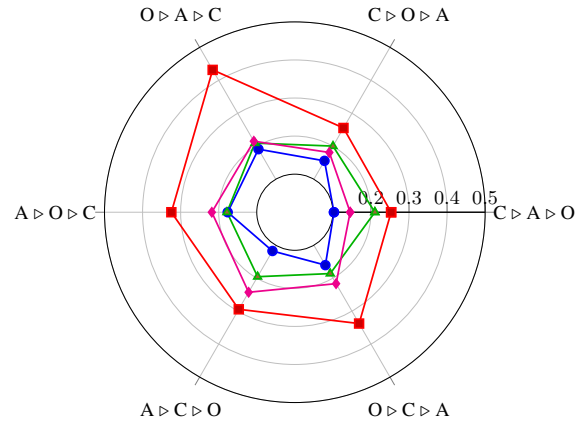


Figure 8: Radar plot comparing F-scores across different datasets and element orders for the Llama 3B model.

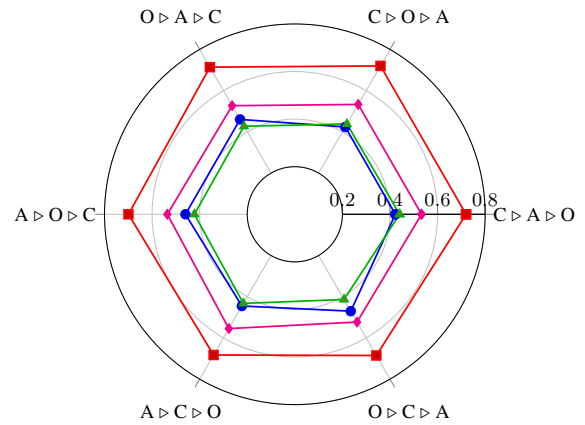


Figure 9: Radar plot comparing F-scores across different datasets and element orders for the Llama 70B model.



Datasets	Laptop 16	Restaurant 16	MAMS	Shoes
N. train samples	2468	1954	3149	906
N. val samples	n/a	n/a	400	116
N. test samples	579	571	400	125
N. categories	67	12	8	21

Table 2: Dataset statistics for the four datasets employed in our study. The number of instances comes after pre-processing, omitting any examples with conflicting labels. Laptop16 and Restaurant16 do not have a validation dataset.

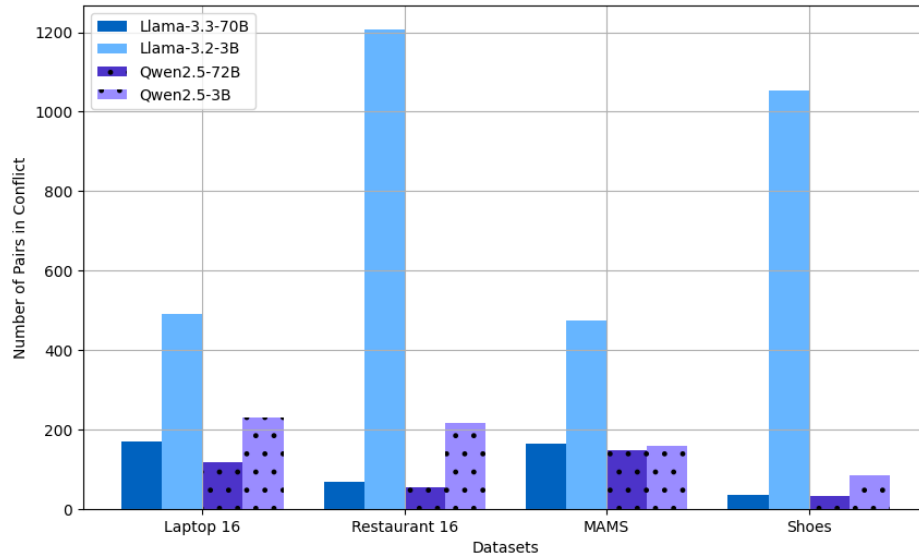


Figure 10: Number of conflicting pairs for each dataset per model for the Joined CoT Agent.

## E CoT Prompt

The prompts of our CoT agents are compositional in the sense that each element has its own template, whether it is the first or a subsequent element in the chain. As previously discussed, we have three element permutations consisting of the aspect term, aspect category, and sentiment polarity. Thus, we have  $3 \times 2 = 6$  templates, which we chain according to the given order to form the CoT agent's prompt. Below, we provide an example of the Aspect  $\rightarrow$  Opinion  $\rightarrow$  Category ordering. As shown, the instructions for each element are numbered and presented sequentially. The prompt for the first element differs in that it includes the review text and begins with the phrase "Given the following text". After these instructions, we prompt the LLM to provide answers by enumerating the elements and including blanks for the LLM to fill in. Finally, we instruct the model to produce the ACSA tuples formatted as a Python list of tuples.

We obtained more stable results by using different types of brackets when specifying the output, i.e., square brackets and parentheses, rather than nested square brackets. The same logic applies to quotation marks: to avoid obfuscating the instructions, we use double quotes around the example demonstration list of tuples and enclose the strings within single quotes.

Additionally, we first present the system instruction used in our study, adapted from (Wang and Luo, 2023), which employs a role-playing format and constrains the LLM to generate the most probable answers while controlling their verbosity.

The fonts in the examples are for illustration purposes only.

### CoT Prompt

1. Given the following text, list all word sequences that denote an aspect term of the restaurant domain:

*"We went again and sat at the bar this time, I had 5 pints of guinness and not one buy-back, I ordered a basket of onion rings and there were about 5 in the basket, the rest was filled with crumbs, the chili was not even edible."*

2. List all word sequences that denote or link to an opinion from the aspects detected.

3. List the categories from the opinions detected. The list of possible categories is: [menu, service, price, ambience, place, staff, miscellaneous, food].

---

1. Aspects:

2. Opinions:

3. Categories:

Lastly, please provide one Python-type list of tuples such as

```
'''[('example\_category_1', 'positive'), ('example\_category_2', 'negative'), ...]'''
```

where the categories are provided above and the sentiment is either *positive*, *neutral*, or *negative*, based on the extracted opinions.

### System Instruction

You are a Natural Language Processing assistant, expert in Aspect-Based Sentiment Analysis. I want you to force yourself to pick words that you are being asked and only them, without explanations or reasoning. If you are unsure, put the most probable. Now follow the following steps:

## F Multi-Hop CoT Prompt

To compare with previous CoT research in the ABSA domain, we adapted the multi-hop approach of Fei et al. (2023); Wang and Luo (2023) to our ACSA context. Unlike Fei et al. (2023), who first provide the aspect term to search, then ask the LLM to identify it from the text, and subsequently use this answer to extract the opinion from the text, which they finally feed into the last prompt to determine the sentiment polarity, we do not assume prior knowledge of the targets. Instead, we follow the same sequential approach as illustrated in our examples in Appendix E, but in a multi-hop manner. After each step, we pose a question to the LLM, and the retrieved answer is added to the message history using the chat template. Below, we demonstrate an example using the 3B Llama model as the assistant, which ultimately provides the ACSA tuples as a Python-style list of tuples for post-processing. Consequently, we make four calls to the LLM for one prediction. Additionally, we would use the same system instruction as in Appendix E.

**CoT Multi-Hop Thread**

**User:**  
Given the following text, list all word sequences that denote an aspect term of the restaurant domain:  
*"We went again and sat at the bar this time, I had 5 pints of guinness and not one buy-back, I ordered a basket of onion rings and there were about 5 in the basket, the rest was filled with crumbs, the chili was not even edible."*

**Assistant:**  
bar, onion, rings, chili

**User:**  
List all word sequences that denote or link to an opinion from the aspects detected.

**Assistant:**  
not, even, edible

**User:**  
List the categories from the opinions detected. The list of possible categories is:  
['menu', 'service', 'price', 'ambience', 'place', 'staff', 'miscellaneous', 'food']

**Assistant:**  
['menu', 'food']

**User:**  
Lastly, please provide one Python type list of tuples such as:  
[('example\_category\_1', 'positive'), ('example\_category\_2', 'negative'), ...]  
that you identified. Where the categories are provided above and the sentiment is either 'positive', 'neutral' or 'negative', based on the extracted opinions.

**Assistant:**  
[('menu', 'negative'), ('food', 'negative')]

## G Few Shot CoT

The prompt used to construct the few-shot CoT prompt, incorporating ten examples from the training dataset, required careful engineering. Within the CoT, the elements—aspect terms, opinions, and categories—are enumerated in varying orders. However, since the datasets employed in this study are ACSA-type, the ground-truth annotations con-

tain only categories and polarities without explicit annotations for aspect terms and opinions. Consequently, the prompt was designed to explicitly illustrate the internal CoT reasoning steps that the LLM should follow between clearly defined BEGIN COT and END COT markers, subsequently instructing the model to output the final result explicitly as a Python list. Given the complexity and detailed nature of this prompt, additional precision was required, including explicit reminders to output the Python list after completing the CoT reasoning steps; this explicit instruction was deemed particularly necessary for the 3B parameter models. Below is a prompt containing two examples.

#### Few Shot CoT Prompt (1/2)

Given the following text, list all word sequences that denote an aspect term of the restaurant domain:

*"Worst Service I Ever Had"*

List all word sequences that denote or link to an opinion from the aspects detected.

List the categories from the opinions detected. The list of possible categories is:

```
[('FOOD#QUALITY', 'AMBIENCE#GENERAL',
'SERVICE#GENERAL', 'RESTAURANT#PRICES',
'DRINKS#QUALITY', 'FOOD#PRICES',
'RESTAURANT#MISCELLANEOUS', 'LOCATION#GENERAL',
'DRINKS#STYLE_OPTIONS', 'DRINKS#PRICES',
'FOOD#STYLE_OPTIONS', 'RESTAURANT#GENERAL'].
```

- The reasoning must appear only between BEGIN COT and END COT.

BEGIN COT

Aspects: ...

Opinions: ...

Categories: ...

END COT

- Outside those markers print one PYTHON LIST of tuples, exactly like

```
[('example_category_1', 'negative'),
('example_category_2', 'positive')] that you identify in your three step COT reasoning.
```

- Where the categories derive from step 3 Categories in COT and each associated category's sentiment is either 'positive', 'neutral' or 'negative', based on step 2 the extracted Opinions.

#### Few Shot CoT Prompt (2/2)

##### EXAMPLES

Example 1

Review: *"Service was wonderful;"*

BEGIN COT

Aspects: ...

Opinions: ...

Categories: ...

END COT

PYTHON LIST: [('SERVICE#GENERAL', 'positive')]

Example 2

Review: *"My mom originally introduced me to this place, but even she (being Indian) feels the food can be somewhat over the top spicy and far too oily."*

BEGIN COT

Aspects: ...

Opinions: ...

Categories: ...

END COT

PYTHON LIST: [('FOOD#QUALITY', 'negative')]

##### NOW SOLVE THE NEW REVIEW

- Please complete the below CoT and end it with END COT.

Given the following text, list all word sequences that denote an aspect term of the restaurant domain:

*"Worst Service I Ever Had"*

List all word sequences that denote or link to an opinion from the aspects detected.

List the categories from the opinions detected. The list of possible categories is:

```
[('FOOD#QUALITY', 'AMBIENCE#GENERAL',
'SERVICE#GENERAL', 'RESTAURANT#PRICES',
'DRINKS#QUALITY', 'FOOD#PRICES',
'RESTAURANT#MISCELLANEOUS', 'LOCATION#GENERAL',
'DRINKS#STYLE_OPTIONS', 'DRINKS#PRICES',
'FOOD#STYLE_OPTIONS', 'RESTAURANT#GENERAL'].
```

- After completing the three step COT reasoning and closing it with END COT, print the PYTHON LIST of tuples, exactly like [('example\_category\_1', 'negative'), ('example\_category\_2', 'positive')] that you identify in your three step COT reasoning.

- Where the categories derive from step 3 Categories in COT and each associated category's sentiment is either 'positive', 'neutral' or 'negative', based on step 2 the extracted Opinions.

- DO NOT FORGET the PYTHON LIST.

BEGIN COT

Aspects: ...



## H CoT versus Multi-Hop CoT

We conducted experiments using the 3B LLMs to determine whether enumerating tasks within a single CoT prompt (see example in Appendix E) or using four separate calls via the multi-hop CoT approach (see example in Appendix F) yields higher prediction scores. Table 3 presents the best-performing CoT agent for each approach, organised by LLM and dataset. Recall that we have six different agents, each varying the order of elements. As shown in the table, one notable characteristic of the multi-hop CoT approach is that Opinions always precede Categories and Aspect terms.

Datasets	Laptop 16		Restaurant16		MAMS		Shoes	
3B LLM models	Qwen	Llama	Qwen	Llama	Qwen	Llama	Qwen	Llama
Enumerated CoT	O \ C \ A: <b>20.3</b>	O \ A \ C: <b>19.1</b>	O \ C \ A: <b>60.0</b>	O \ A \ C: 43.1	A \ C \ O: 26.6	C \ A \ O: 21.0	A \ C \ O: <b>37.1</b>	A \ C \ O: 24.2
Multi-hop CoT	O \ C \ A: 15.9	O \ C \ A: 17.1	O \ C \ A: 56.5	O \ C \ A: <b>52.5</b>	O \ C \ A: <b>28.1</b>	O \ C \ A: <b>29.1</b>	O \ C \ A: 29.2	O \ A \ C: <b>29.2</b>

Table 3: The enumerated single prompt CoT against the multi-hop CoT. The table shows the F1 score in percentages across the four datasets and the 3B Qwen and Llama models. The O, C, and A stand for Opinion, Category and Aspect terms accordingly. They show the order of the elements from left to right. Each cell has the top performant element order measured for that dataset and LLM. On the right of the element order is depicted the F1 score, bold shows the higher percentage between Enumerated CoT and Multi-hop CoT.