Multi-level Association Refinement Network for Dialogue Aspect-based Sentiment Quadruple Analysis

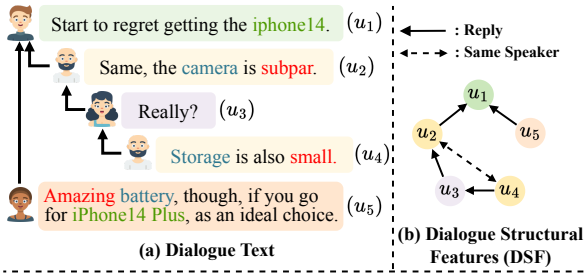
Zeliang Tong[1], Wei Wei[1†], Xiaoye Qu[1], Rikui Huang[1], Zhixin Chen[2], Xingyu Yan[3]
[1]Cognitive Computing and Intelligent Information Processing (CCIIP) Laboratory,
School of Computer Science and Technology, Huazhong University of Science and Technology
[2]Shenzhen Yishi Huolala Technology Limited [3]State Grid Fujian Electric Power Co.
tongzeliang744@gmail.com, weiw@hust.edu.cn

Abstract

Dialogue Aspect-based Sentiment Quadruple (DiaASQ) analysis aims to identify all quadruples (*i.e.*, <target, aspect, opinion, sentiment>) from the dialogue. This task is challenging as different elements within a quadruple may manifest in different utterances, requiring precise handling of associations at both the utterance and word levels. However, most existing methods tackling it predominantly leverage predefined dialogue structure (*e.g.*, reply) and word semantics, resulting in a surficial understanding of the deep sentiment association between utterances and words. In this paper, we propose a novel **M**ulti-level **A**ssociation **R**efinement **N**etwork (MARN) designed to achieve more accurate and comprehensive sentiment associations between utterances and words. Specifically, for utterances, we dynamically capture their associations with enriched semantic features through a holistic understanding of the dialogue, aligning them more closely with sentiment associations within elements in quadruples. For words, we develop a novel cross-utterance syntax parser (CU-Parser) that fully exploits syntactic information to enhance the association between word pairs within and across utterances. Moreover, to address the scarcity of labeled data in DiaASQ, we further introduce a multi-view data augmentation strategy to enhance the performance of MARN under low-resource conditions. Experimental results demonstrate that MARN achieves state-of-the-art performance and maintains robustness even under low-resource conditions.

1 Introduction

In recent years, Aspect-based Sentiment Analysis (ABSA) has become a key research focus, benefiting downstream tasks like emotional conversation generation (Zhao et al., 2023; Wang et al., 2023; Lu et al., 2023) and recommendation systems (Wei et al., 2019; Liu et al., 2022; Zou et al., 2024). However, much of this research focuses on isolated texts, limiting its broader use, especially in multi-turn dialogues on real-world social media platforms (*e.g.*, Twitter, Facebook, TikTok). To overcome this challenge, Li et al. (2023a) pioneered the development of Dialogue Aspect-based Sentiment Quadruple (DiaASQ) analysis, which extends single text-level ABSA to a dialogue context. As shown in Figure 1, DiaASQ aims to identify all sentiment quadruples within a given dialogue, encompassing the *target*, associated *aspect*, *opinion*, and *sentiment polarity*, which may appear across various utterances.



(a) Dialogue Text

(b) Dialogue Structural Features (DSF)

Target	*Aspect*	*Opinion*	*Sentiment*	*Utterances*	*Speaker*
iphone14	camera	subpar	negative	$\{u_1, u_2\}$	{ , }
iphone14	storage	small	negative	$\{u_1, u_4\}$	{ , }
iphone14 Plus	battery	amazing	positive	$\{u_5\}$	{ }

(c) Sentiment quadruples extracted from the dialogue

Figure 1: Examples of DiaASQ task from the real-world dataset. "Target", "Aspect", and "Opinion" are highlighted in green, blue, and red, respectively. "u_i" denotes an utterance, while the avatar icons and arrows represent the speakers and reply relationships.

Unlike single sentence-based ABSA, DiaASQ encounters markedly greater challenges due to the dispersion of sentiment elements across diverse utterances, necessitating a meticulous grasp of the intricate associations between utterances and words. To address this, existing frameworks (Li et al., 2023a, 2024a,b; Zhou et al., 2024) often exploit predefined dialogue structural features (DSF) shown Figure 1 (b) to determine utterance-level associations and leverage word semantics to capture word-

† Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14035–14057

level associations. Despite their success, they still face substantial limitations. **From an utterance-level perspective**, *irrelevant* and *implicit* associations brought by seemingly intuitive DSF often distort quadruple extraction. For example, as shown in Figure 1 (a), although u_5 is a response to u_1, the discussion topics ("iPhone 14" and "iPhone 14 Plus") are entirely different (*irrelevant*). While u_1 and u_4 share the sentiment target "iPhone 14" but do not have a direct DSF that emphasizes their association (*implicit*). Therefore, while DSF modestly provides clues about sentiment associations between utterances, its inherent limitations compel existing solidified paradigms that solely rely on DSF, failing to filter out irrelevant associated utterances and overlooking broader utterance associations beyond predefined relationships. **From a word-level perspective**, the relatively extended length of dialogue texts complicates the establishment of long-range associations between words (Liu et al., 2024). Unfortunately, previous studies overlook the effective construction of cross-utterance syntax information between words in complex dialogues, which stays valid in the long-distance dependency problem.

Moreover, in practical applications, the DiaASQ task faces significant challenges in acquiring large-scale annotated datasets due to both the difficulty and cost involved. Li et al. (2023a) successfully curated a high-quality set of 1,000 tree-structured dialogues from an initial pool of approximately 9 million dialogues, following a rigorous screening process (*e.g.*, abusive filtering, ethical reviews) and fine-grained annotation. Consequently, it is crucial to explore strategies for maintaining model performance under conditions of limited labeled data.

To solve the above-mentioned issues and challenges, we propose a novel Multi-level Association Refinement Network (MARN). **For the utterance level**, we propose a DSF Optimization Module (DOM) to refine the inherently flawed utterance-level associations provided by DSF, aligning them more closely with sentiment associations. Specifically, DOM enriches utterance representations through a thorough understanding of the overarching dialogue semantics, autonomously filtering out explicit associations with sentiment-irrelevant utterances in DSF while explicitly modeling the implicit associations between sentiment-relevant utterances. **For the word level**, we develop a novel Cross-Utterance Syntax Parser (CU-Parser) that establishes syntax relations for word pairs not only within but also across utterances. Given their proficiency in tracking and integrating non-sequential syntactic relationships (Sun et al., 2019; Tian et al., 2021b; Liang et al., 2022), the application of CU-Parser in dialogues enhances the associations between words and improves the mutual perception of sentiment elements, particularly for those exhibiting long-range dependencies across disparate utterances. **Additionally**, to mitigate the constraint imposed by limited labeled data in DiaASQ, we develop a cost-effective multi-view data augmentation scheme to address low-resource scenarios without extensive manual fine-grained annotations.

Our main contributions are as follows:

(1) We identified the inherent flaws in previous works, specifically the solidified handling of DSF and insufficient consideration of word dependencies, leading to irrelevant and implicit associations between utterances and weak word associations.

(2) We propose a novel Multi-level Association Refinement Network (MARN). Specifically, it employs the DOM to dynamically capture the sentiment associations between utterances and utilizes a carefully designed CU-Parser to establish syntax associations between words both across and within utterances. Furthermore, it incorporates a multi-view data augmentation strategy to alleviate the challenges posed by limited resources.

(3) Experimental results show that MARN surpasses previous state-of-the-art methods on benchmark datasets, while consistently maintaining robust performance in low-resource settings.

2 Related Works

2.1 Aspect-based Sentiment Analysis

Aspect-based Sentiment Analysis (ABSA) primarily forecasts various sentiment elements or combinations. The most three basic subtasks are Aspect Term Extraction (ATE) (Ma et al., 2019; Yang et al., 2020), Opinion Term Extraction (OTE) (Wan et al., 2020; Veyseh et al., 2020) and Aspect Sentiment Classification (ASC) (Tian et al., 2021a; Wang et al., 2021; Zhou et al., 2021). Recently, research has shifted to compound tasks like aspect-based sentiment triplet or quadruple extraction, involving multiple sentiment elements. Some milestone solution paradigms have emerged, such as table-tagging methods (Wu et al., 2020; Chen et al., 2022a), span-based methods (Xu et al., 2021; Chen et al., 2022b), machine reading comprehension-based method (Zhang et al., 2020a; Mao et al., 2021) and generative-based method (Zhang et al.,

2021; Gou et al., 2023). However, renowned ABSA benchmarks such as SemEval (Pontiki et al., 2016), MAMS (Jiang et al., 2019), and Twitter (Dong et al., 2014) are annotated solely at the single sentence level, limiting the seamless adaptation of existing frameworks to the broader and more practically significant multi-utterance dialogue scenarios.

To address the above limitations, Li et al. (2023a) first introduced the DiaASQ task and proposed a baseline model that leverages predefined DSF (*i.e.*, replies, by the same speaker, in the same thread) to manage utterance associations through three parallel attention matrices. Building on this, subsequent works have adopted similar approaches to DSF usage while introducing optimizations and innovations from perspectives such as initial encoding (Li et al., 2024b), final decoding (Li et al., 2024a), and efficiency (Zhou et al., 2024). However, these approaches overlook the inherent incompleteness of DSF, making them susceptible to irrelevant information and incapable of deeply exploring implicit associations. To tackle these issues, this paper proposes a novel DOM to refine DSF, enabling comprehensive and precise information aggregation.

2.2 Syntactic Dependency Analysis

Syntactic Dependency Analysis (SDA) aims to exploit the linguistic features of a sentence by identifying the dependencies between words (Gu et al., 2022; Xiao et al., 2020; Zhu et al., 2023). By leveraging such non-sequential information, SDA effectively shortens paths between syntactically related words, provides an essential complement to semantic features for long-text understanding, thus remains valid in downstream tasks involving long-distance dependency problems, like ABSA (Liang et al., 2022; Chen et al., 2022a, 2021), Relation Extraction (Tian et al., 2021c, 2022; Cheng et al., 2021; Zhu et al., 2021). However, traditional SDA is constrained to single-sentence and fails to capture word associations across utterances, which motivates us to propose a novel CU-Parser to establish more effective connections between sentiment elements across utterances.

3 Preliminary

3.1 Problem Statement

Given the input dialogue $\mathcal{D} = \{(u_i, s_i, r_i)\}_n$, where n is the number of utterances contained in the dialogue. Here, $u_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$ denotes the i-th utterance consisting of tokens w_{ij}. The variable s_i denotes the speaker associated with u_i and r_i signifies that the i-th utterance responds to the r_i-th utterance. The goal of DiaASQ task is to extract all target-aspect-opinion-sentiment quadruples, denoted as $\mathcal{Q} = \{(t, a, o, p)_q\}_{q=1}^{|\mathcal{Q}|}$. Here, t, a, and o represent the target, aspect, and opinion found within the dialogue, respectively, and p represents sentiment polarity in $\{pos, neg, neu, other\}$.

3.2 Dialogue Structural Features

Following the previous research (Li et al., 2024b), to represent the initial dialogue structural features, we define two adjacency sets for each utterance: $\mathcal{R}$ and $\mathcal{S}$. Here, $\mathcal{R}_i$ denotes the set of indices of utterances that reply to u_i, and $\mathcal{S}_i$ represents the set of indices of utterances from the same speaker as u_i, formally expressed as:

$$\mathcal{R}_i = \{k \mid r_k = i\}, \ \mathcal{S}_i = \{j \mid s_j = s_i\}. \quad (1)$$

We define matrix $\mathbf{A}$ to denote the initial association between utterances:

$$\mathbf{A}_{ij} = \begin{cases} \frac{1}{|\mathcal{R}_i|+|\mathcal{S}_i|+1} & \text{if } j \in \mathcal{R}_i \cup \mathcal{S}_i \text{ or } i = j \\ 0 & \text{Otherwise} \end{cases}. \quad (2)$$

4 Methodology

This section comprehensively explains our proposed MARN, as illustrated in Figure 2. The encoder first generates contextualized representations. Subsequently, the DSF Optimization Module (DOM) refines the associations between utterances, transforming the original DSF into DSF′ to better reveal sentiment associations. Feature aggregation is then conducted by incorporating both utterance-level associations and word-level semantic and syntax associations. Finally, MARN decodes all quadruples using a grid-tagging scheme.

4.1 Dialogue Encoder

We leverage a Pretrained Language Model (PLM) to obtain contextualized word embeddings. For utterances in $\mathcal{D}$, we concatenate them sequentially, and inserting [SEP] to delimit adjacent sentences:

$$\begin{aligned} \mathbf{H} &= \text{PLM}\left(\{u_1, [\text{SEP}], u_2, \ldots, u_n\}\right), \\ &= [\{\mathbf{h}_{1j}\}, \mathbf{h}_{sep}, \{\mathbf{h}_{2j}\}, \ldots, \{\mathbf{h}_{nj}\}], \end{aligned} \quad (3)$$

where $\mathbf{h}_{ij}$ is the contextual embedding of the j-th word in the i-th utterance.

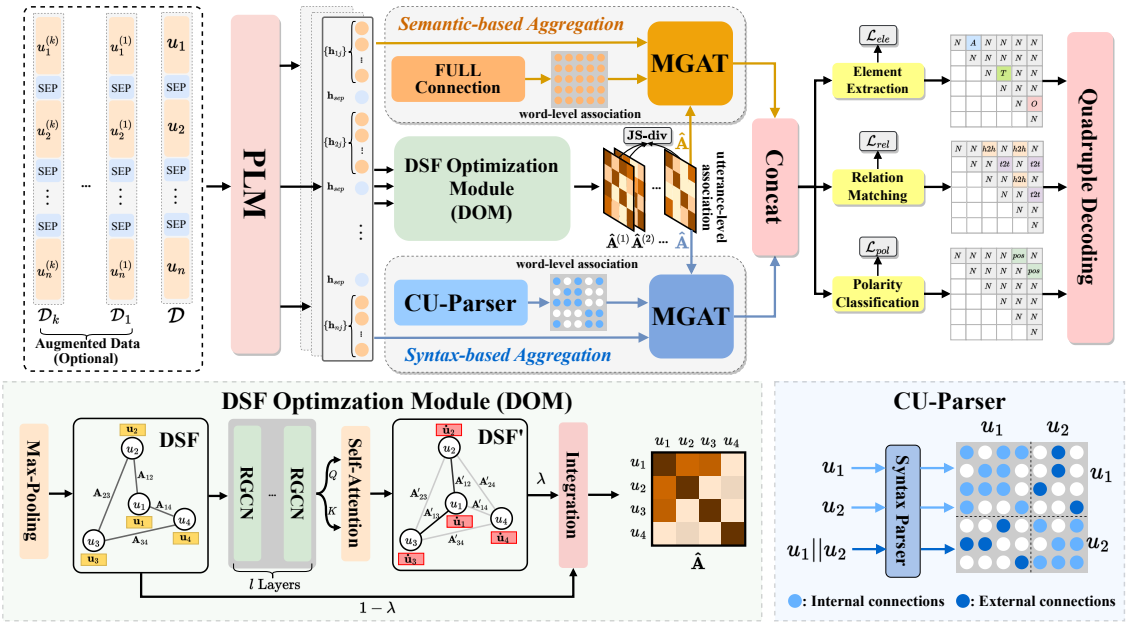


Figure 2: Overall architecture. First, PLM is used to generate contextual representations. Next, DOM refines the original DSF to obtain refined utterance association, while CU-Parser provides refined cross-utterance syntax. Subsequently, MGAT integrates the refined utterance and word level associations, performing feature aggregation in parallel based on semantic and syntactic information. Finally, MARN decodes quadruples based on grid-tagging.

4.2 DSF Optimization Module

The DSF Optimization Module (DOM) is designed to dynamically refine utterance associations, ensuring closer alignment with the conditions necessary for sentiment quadruple extraction through a holistic understanding of the dialogue. First, we form the utterances representations by:

$$\mathbf{U} = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_n}], \tag{4}$$

$$\mathbf{u_i} = \text{Max-Pooling}\left(\{\mathbf{h}_{ij}\}_{j=1}^{|u_i|}\right). \tag{5}$$

We now update and enrich the representation for each utterance to achieve a comprehensive understanding of the dialogue using the Relation Graph Convolutional Network (RGCN) as follows:

$$\mathbf{u}_i^l = \sigma\Big(\sum_{r\in\mathcal{F}}\sum_{j\in r_i}\frac{1}{|r_i|}\mathbf{W}_r^l\mathbf{u}_j^{l-1} + \mathbf{W}_0^l\mathbf{u}_i^{l-1}\Big). \tag{6}$$

Here, $\mathcal{F} = \{\mathcal{R}, \mathcal{S}\}$, and the output of the last layer of RGCN is defined as $\hat{\mathbf{u}}$ for simplicity. We next compute the association of each utterance pair in DSF$'$ by the following self-attention mechanism:

$$\mathbf{A}' = \text{Softmax}(\mathbf{W}_q\hat{\mathbf{U}}(\mathbf{W}_k\hat{\mathbf{U}})^\top/\sqrt{d}), \tag{7}$$

where $\mathbf{A}'$ represents the associations between utterances in DSF$'$. Finally, we combine the original DSF with the optimized version DSF$'$ as follows:

$$\hat{\mathbf{A}} = (1-\lambda)\mathbf{A} + \lambda\mathbf{A}', \tag{8}$$

here, λ represents the optimized confidence weight.

4.3 CU-Parser

To refine the word-level associations and strengthen the perceptual connections between sentiment elements both within and across utterances, we propose the CU-Parser. This advanced framework extends conventional single-sentence syntactic dependency parsing to the more complex multi-utterance dialogues, as demonstrated in Figure 2.

- **Internal Connection**. Each single utterance is parsed independently by the syntax parser.
- **External Connection**. For any two utterances, we concatenate them using the conjunction "and", which is then processed by the syntax parser and preserves the inter-sentential connections as the syntax link between utterances. We also proposed other semantic-based connection methods, detailed in Appendix E.

We denote $\mathcal{N}_{ij}^{syn}$ as words directly syntactically connected to w_{ij}, which is then combined with the DOM output, as discussed in the next subsection.

4.4 Semantic and Syntax-based Aggregation

Upon refining associations at the utterance and word levels, we propose a Multi-level Graph Attention Network (MGAT) to combine these multi-level associations for aggregating information-rich and concise word representations, formulated as:

$$\mathbf{h}_{ab}^l = \delta\Big(\sum_{cd\in\mathcal{N}_{ab}}\alpha_{ab,cd}^l\mathbf{W}_g^l\mathbf{h}_{cd}^{l-1}\Big), \tag{9}$$

$$\alpha^{l}_{ab,cd} = \frac{\hat{\mathbf{A}}_{ac} \cdot e^{f\left(\mathbf{h}^{l-1}_{ab}, \mathbf{h}^{l-1}_{cd}\right)}}{\sum_{c'd' \in \mathcal{N}_{ab}} \hat{\mathbf{A}}_{ac'} \cdot e^{f\left(\mathbf{h}^{l-1}_{ab}, \mathbf{h}^{l-1}_{c'd'}\right)}}, \quad (10)$$

where $\alpha_{ab,cd}$ carries the multi-view association between words w_{ab} and w_{cd}, with $\hat{\mathbf{A}}_{ac}$ representing the utterance-level association between u_a and u_c, and f capturing the word-level association, δ is an activation function. Final output is $\hat{\mathbf{h}}$ for simplicity.

This aggregation process is applied twice: in the semantic-based aggregation, $\mathcal{N}$ encompasses all words in the dialogue, whereas in the syntax-based aggregation, $\mathcal{N}$ is initialized by $\mathcal{N}^{syn}$. Finally, we form the final token representation by,

$$\mathbf{s}_{ij} = \hat{\mathbf{h}}^{sem}_{ij} \oplus \hat{\mathbf{h}}^{syn}_{ij}, \quad (11)$$

where $\oplus$ means vector concatenation.

4.5 Multi-view Data Augmentation

To address the resource constraints in the DiaASQ task, we propose a cost-effective multi-view data augmentation approach that enhances overall performance by improving the robustness of DSF optimization. This method automatically applies simple semantic similarity transformations to the original data from two perspectives:

- **Word Level.** To mitigate the overfitting of specific words, we prompt large language models (LLMs) to rephrase key backbone words of part of utterances with synonyms.
- **Utterance Level.** To mitigate the overfitting of specific utterance expressions, we prompt LLMs to rephrase part of the utterances in the dialogue, preserving the overall semantic meaning while varying the expressions.

We denote the augmented dataset for each dialogue as $\mathcal{U} = \{\mathcal{D}_i\}_k$. Note that $\mathcal{U}$ is solely used for optimizing DSF to prevent excessive training time. Further examples and detailed explanations of the data augmentation process can be found in Appendix H. Similarly, we use DOM to obtain the inter-utterance associations for each $\mathcal{D}_i$:

$$\mathcal{A}ug = [\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}, \ldots, \hat{\mathbf{A}}^{(k)}]. \quad (12)$$

To ensure the model robustly optimizes DSF across similar expressions, we introduce a diversity loss based on the Jensen-Shannon (JS) divergence, which minimizes the discrepancies in utterance associations across semantically similar expressions:

$$\mathcal{L}_{JS\text{-}div} = \frac{1}{k} \sum_{i=1}^{k} \text{JS}\left(\hat{\mathbf{A}} || \hat{\mathbf{A}}^{(i)}\right). \quad (13)$$

By minimizing the JS divergence, the model can avoid over-reliance on specific expressions, ensuring the robustness of the DSF optimization process.

It is worth noting that this component is optional, and removing the data augmentation part does not impact the progression of MARN.

4.6 Quadruple Decoding

After obtaining the final word representations, we streamline quadruple extraction by decomposing the decoding task into three subtasks: sentiment element extraction (*ele*), relation matching (*rel*), and sentiment polarity classification (*pol*). The word pair label set for each subtask is denoted as $tag^{\mathcal{C}}$, where $\mathcal{C} \in \{ele, rel, pol\}$.

- **Element**. $tag^{ele} \in \{T, A, O, N\}$, where T, A, and O denote the first and last words of a valid target, aspect, and opinion, respectively, and N represents an invalid word pair.
- **Relation**. $tag^{rel} \in \{h2h, t2t, N\}$, where $h2h$ denotes a headword pair within a matching sentiment element pair, and $t2t$ marks a tail word pair. For example, for the aspect-opinion pair (storage capacity, not sufficient), the head words "storage" and "not" are labeled as $h2h$, while the tail words "capacity" and "sufficient" are labeled as $t2t$.
- **Polarity**. $tag^{pol} \in \{Pos, Neg, Neu, N\}$, where Pos, Neg, and Neu indicate word pairs with positive, negative, or neutral polarities, and N denotes an invalid word pair.

For each subtask, we use the grid labeling method to classify each word pair, which is formulated as,

$$\mathbf{v}_{ab,cd} = \mathbf{s}_{ab} \oplus \mathbf{s}_{cd}, \quad (14)$$

where $\oplus$ represents the vector concatenation. This is then passed through a fully connected layer with a softmax activation function to produce the probability distribution for each label in each subtask.

$$\mathbf{p}^{\mathcal{C}}_{ab,cd} = \text{Softmax}\left(\mathbf{W}_c \mathbf{v}_{ab,cd} + b_c\right), \quad (15)$$

where $\mathcal{C} \in \{ele, rel, pol\}$ represents the set of tasks, and $\mathbf{W}_c$ and $\mathbf{b}_c$ are trainable parameters. We combine the predicted labels of the three subtasks to output quadruples. Appendix D gives a more detailed decoding process.

4.7 Training Objective

The final loss of each subtask is defined as the cross-entropy loss between the ground truth labels and the predicted distributions for all word pairs:

$$\mathcal{L}_{\mathcal{C}} = -\frac{1}{N^2}\sum_{ab}\sum_{cd}\mathbf{y}^{\mathcal{C}}_{ab,cd}\log(\mathbf{p}^{\mathcal{C}}_{ab,cd}), \quad (16)$$

where N is the total number of words in the dialogue, $\mathbf{y}^{\mathcal{C}}_{ab,cd}$ and $\mathbf{p}^{\mathcal{C}}_{ab,cd}$ is the ground-truth and prediction of the corresponding subtask $\mathcal{C}$. The overall loss $\mathcal{L}$ is computed as the weighted sum of the losses for all subtasks and JS-divergence loss:

$$\mathcal{L} = \mathcal{L}_{ele} + \beta\mathcal{L}_{rel} + \eta\mathcal{L}_{pol} + \gamma\mathcal{L}_{JS\text{-}div}. \quad (17)$$

Here, $\mathcal{L}_{ele}$, $\mathcal{L}_{rel}$, and $\mathcal{L}_{pol}$ denote the losses for the three subtasks, respectively.

5 Experiment

5.1 Dataset and Implementation Detail

We conducted experiments on English and Chinese datasets from Li et al. (2023a) to evaluate our model, focusing on the mobile phone domain. Each data point consists of an initial post followed by sequential responses from multiple individuals.

Following the previous studies, we employ RoBERTa-Large (Liu et al., 2019) for the English dataset and Chinese-RoBERTa-wwm-ext (Cui et al., 2020) for the Chinese dataset to initialize our PLMs. SuPar (Zhang et al., 2020b) is adopted as the syntax parser in CU-Parser. "Micro F1" is the evaluation metric consistent with prior work. Appendix A details the datasets and implementation. We also provide efficiency analysis in Appendix B.

5.2 Baselines

We compare MARN with the following baselines, which can be briefly grouped into three categories:

- **ABSA Baselines**. **EC-ACOS** (Cai et al., 2021) first extracts aspect-opinion pairs and then predicts category-sentiment. **SpERT** (Eberts and Ulges, 2020) is a span-based model for joint entity and relation extraction. **Span-ASTE** (Xu et al., 2021) captures span-to-span interactions for relation extraction. **ParaPhrase** (Zhang et al., 2021) generatively predicts sentiment quadruples in one step.
- **DiaASQ Baselines**. **Meta-WP** (Li et al., 2023a), **H2DT** (Li et al., 2024a), **STS** (Zhou et al., 2024) and **DMCA** (Li et al., 2024b) incorporate prior predefined DSF and word semantics to extract sentiment quadruples.
- **LLMs**. Given the advancement of large language models (LLMs) in various tasks (Xie et al., 2024; Zhang et al., 2024), we employed ChatGPT[1] with in-context learning (ICL) as baselines. We also deployed other series and fine-tuned LLMs, detailed in the Appendix G.

[1]We used the ChatGPT model `gpt-4-turbo-2024-04-09` in our experiments.

5.3 Main Results

Table 1 presents the main results. Observations are:

(1) Our MARN model outperforms all baselines, confirming its superiority. It introduces an effective multi-level association refinement method for the DiaASQ task, enabling better use of both utterance-level DSF and word-level syntax for selecting sentiment-related information.

(2) Overall, the DiaASQ baseline utilizing the predefined DSF surpasses the ABSA baseline without DSF. This observation suggests that DSF offers valuable cues for sentiment quadruple extraction. However, the solidified application of DSF, along with its inherent irrelevant and implicit association limitations, constrains further performance gains.

(3) MARN shows the most significant performance improvement in compound subtasks (*e.g.*, $\mathcal{T}$-$\mathcal{A}$, $\mathcal{T}$-$\mathcal{O}$, $\mathcal{A}$-$\mathcal{O}$, $\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$) of DiaASQ. While the model provides only modest improvements in the extraction of individual sentiment elements, it significantly enhances the matching performance between elements, thereby improving the overall accuracy of sentiment quadruple extraction.

(4) Although LLMs can perform few-shot inference, they perform much worse than fine-tuned small PLMs, suggesting that LLMs like ChatGPT may not be suitable for complex DiaASQ tasks. We believe this is due to natural language prompts not effectively incorporating DSF, which limits their ability to capture contextual dependencies. Appendix G presents additional experimental results and analysis for various and fine-tuned LLMs, including LLaMa, Qwen, and Mistral.

5.4 Ablation Study

We conduct an ablation study (Table 2) to evaluate model components. Observations are:

(1) **MARN w/o DSF**, which removes the initial DSF connection from $\hat{\mathbf{A}}$. This leads to a 4.99% and 4.80% decrease in F1 for quadruple extraction on the two datasets. This result suggests that the original DSF provides useful cues for sentiment element matching, especially for sentiment elements across utterances with explicit associations.

(2) **MARN w/o DSF′**, which removes the optimized DSF′ connection from $\hat{\mathbf{A}}$. This leads to a 2.96% and 2.81% decrease in F1 for quadruple extraction on the two datasets. This proves that the optimized DSF′ refines and complements the

Model	English								Chinese							
	$\mathcal{T}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$	$\mathcal{T}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$
EC-ACOS (2021)	88.31	71.71	47.90	34.31	20.94	19.21	12.80	11.59	91.11	75.24	50.06	32.47	26.78	18.90	9.25	8.81
SpERT (2020)	87.82	74.65	54.17	28.33	21.39	23.64	13.38	13.07	90.69	76.81	54.06	38.05	31.28	21.89	14.19	13.00
Span-ASTE (2021)	/	/	/	42.19	30.44	45.90	28.34	26.99	/	/	/	44.13	34.46	32.21	30.85	27.42
ParaPhrase (2021)	/	/	/	37.22	32.19	30.78	26.76	24.54	/	/	/	37.81	34.32	27.76	27.98	23.27
Meta-WP (2023a)	**89.56**	74.56	60.06	47.91	45.58	44.27	37.85	33.72	91.79	**78.92**	59.15	48.61	43.31	45.44	38.78	35.21
STS (2024)	89.00	75.09	**63.57**	55.12	53.11	56.52	47.61	43.80	91.49	77.10	61.24	53.56	50.29	53.26	42.82	40.59
DMCA (2024b)	88.11	73.95	63.47	53.08	50.99	52.40	41.00	37.96	92.03	77.07	60.27	56.88	51.70	52.80	45.36	42.68
H2DT (2024a)	88.69	73.81	62.61	48.69	48.84	52.47	42.19	39.01	91.72	76.93	61.87	50.48	48.80	52.40	42.81	40.34
ChatGPT (0-shot)	48.62	33.71	42.25	23.68	21.11	22.78	18.88	16.21	34.32	34.90	45.12	28.01	28.24	24.87	20.04	17.19
ChatGPT (5-shot)	52.54	39.93	47.56	30.29	26.65	24.49	22.45	19.59	44.21	50.62	47.33	30.75	28.29	26.90	22.10	19.76
MARN	89.53	**76.22**	61.75	**55.17**	**55.82***	**58.72***	**48.89**	**44.95***	**92.34**	77.20	**61.94**	**58.02***	**53.55***	**55.24***	**48.23***	**45.41***

Table 1: Main results. "$\mathcal{T}/\mathcal{A}/\mathcal{O}$" represents "target/aspect/opinion" respectively. "$\mathcal{Q}$" represents the sentiment quadruple. Appendix J details the prompt used for ChatGPT. The best results are highlighted in bold, and the second-best results are underlined. All the scores are averaged values over five runs under different random seeds. Significant improvements compared to the best baseline are marked with * (t-test, $p \leq 0.05$).

Model	English		Chinese	
	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$
MARN	**48.89**	**44.95**	**48.23**	**45.41**
- w/o DSF	43.29$_{(\downarrow 5.60)}$	39.97$_{(\downarrow 4.99)}$	42.27$_{(\downarrow 5.96)}$	40.61$_{(\downarrow 4.80)}$
- w/o DSF$'$	43.90$_{(\downarrow 4.98)}$	41.99$_{(\downarrow 2.96)}$	43.87$_{(\downarrow 4.36)}$	42.60$_{(\downarrow 2.81)}$
- w/o CU-Parser	45.01$_{(\downarrow 3.88)}$	41.12$_{(\downarrow 3.83)}$	44.25$_{(\downarrow 3.98)}$	41.70$_{(\downarrow 3.71)}$
- w/o JS-div loss	48.08$_{(\downarrow 0.81)}$	44.45$_{(\downarrow 0.50)}$	47.30$_{(\downarrow 0.93)}$	44.72$_{(\downarrow 0.69)}$
- w/o Aug$_w$	48.24$_{(\downarrow 0.65)}$	44.51$_{(\downarrow 0.44)}$	47.69$_{(\downarrow 0.54)}$	44.79$_{(\downarrow 0.62)}$
- w/o Aug$_u$	48.29$_{(\downarrow 0.70)}$	44.53$_{(\downarrow 0.42)}$	47.49$_{(\downarrow 0.74)}$	44.39$_{(\downarrow 1.02)}$

Table 2: Ablation results. The notation "w/o" signifies excluding the corresponding component from MARN. Aug$_w$ and Aug$_w$ represent the processes of word-level and utterance-level data augmentation.

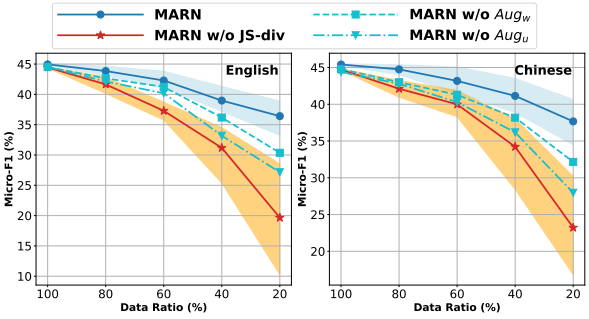


Figure 3: Evaluation results in low-resource scenarios, where the model is trained under different dataset usage ratios from 100% to 20%. The blue and orange shaded areas represent the upper and lower bounds of MARN and MARN w/o JS-div loss. The line represent average results from five runs with different random seeds.

original DSF. Using overall utterance semantics, DSF$'$ mitigates irrelevant associations and captures implicit relationships that DSF fails to model.

(3) **MARN w/o CU-Parser**, which removes the syntax-based aggregation. The removal of cross-utterance syntactic dependency structures leads to performance degradation, as this non-sequential information reveals critical associations between words across and within utterances, which are essential for accurate sentiment element matching.

(4) **MARN w/o JS-div loss, Aug$_w$, and Aug$_u$**, which removes the corresponding augmentation processes. This component provides only modest improvements when trained on the full dataset. In other words, even without data augmentation, MARN remains SOTA and does not introduce additional API calls. Nevertheless, it proves highly effective under low-resource conditions, as detailed in the experiments presented in Section 6.1.

In summary, each module of MARN contributes to the overall performance.

6 Analysis

6.1 Effects of Data Augmentation in Low-Resource Scenario

To further evaluate the effectiveness of the multi-view data augmentation strategy proposed in MARN, we deployed the model under more extreme low-resource conditions. The experimental results are illustrated in Figure 3. Observations are:

(1) Without augmented data, the model's performance declines sharply with smaller datasets and becomes unstable due to data distribution variations. In contrast, data augmentation enables MARN to effectively capture inter-utterance relationships, even with limited resources, easing the challenge of filtering sentiment information in complex dialogues. For example, with only 20% of the dataset, the F1 score for quadruple extraction reached 36% on English and 37% on Chinese datasets. This phenomenon underscores the crit-

ical importance of data augmentation techniques in enhancing model robustness, particularly in natural language processing tasks where contextual understanding is paramount.

(2) Both word-level and utterance-level data augmentation contribute significantly to the overall performance. We infer that this stems from their ability to simulate real-world dialogue scenarios from different perspectives, preventing the model from overfitting specific words or expressions during training. Since the data augmentation is solely designed to reinforce DOM, this indirectly highlights that accurately assessing inter-utterance associations is critical in the DiaASQ task, particularly in low-resource scenarios. More analysis about its effectiveness is given in Appendix H.3.

6.2 Cross Utterance Quadruples Extraction

We evaluated the model's performance in extracting sentiment quadruples across utterances. As shown in Figure 4, we evaluated our MARN model, its variant without the optimized DSF′ and CU-Parser, and the relatively strongest DiaASQ baseline STS. Observations are:

(1) The refinement of associations at the utterance level and the cross-utterance syntactic information at the word level contribute significantly to the extraction of cross-utterance sentiment quadruples. Together, these mechanisms prevent the extraction performance from declining linearly as the cross-utterance level increases, maintaining an F1 score of approximately 30% even at levels ≥ 3. This demonstrates that MARN indeed strengthens the associations between utterances and words. The utterance-level association refinement enables the model to focus on sentiment-relevant utterances by excluding sentiment interference from irrelevant associations and uncovering implicit associations. At the same time, the word-level association refinement strengthens the model's ability to perceive connections between sentiment elements.

(2) Our approach outperforms strong DiaASQ baseline STS in cross-utterance scenarios, with notable gains at higher levels (cross ≥ 3). This underscores the effectiveness of MARN in addressing extraction challenges in multi-turn dialogues. For a more transparent and intuitive illustration, case studies are provided in Appendix I.

6.3 Effects of CU-Parser

We also undertake experiments to investigate various methods for modeling cross-utterance syntax.

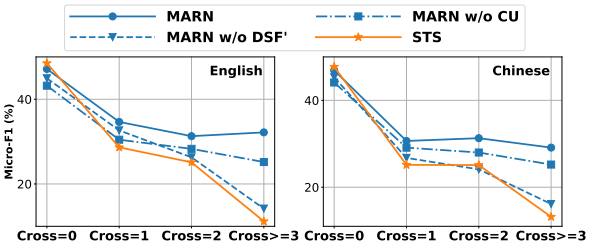


Figure 4: Results of extracting quadruples across different utterances. "Cross" denotes the cross-utterance level of the most distant pair within a quadruple.

Category	Method	English	Chinese
MARN	**CU-Parser**	**44.95**	**45.41**
Connection Word	Root-Root	43.37 (↓1.58)	43.88 (↓1.53)
	V-V	43.34 (↓1.61)	43.62 (↓1.79)
	N-N	42.03 (↓2.92)	43.11 (↓2.30)
	ADJ-ADJ	42.40 (↓2.55)	42.98 (↓2.43)
Connection Type	Internal Only	42.53 (↓2.42)	42.85 (↓2.56)
	External Only	42.65 (↓2.30)	42.83 (↓2.58)
Parser	Stanford Parser	44.39 (↓0.56)	44.71 (↓0.70)

Table 3: Comparison of different methods for constructing cross-utterance syntax. "Root" represents the root word obtained after syntactic dependency parsing.

The results are shown in Table 3. Observations are:

(1) We attempted to heuristically establish cross-utterance connections using other key terms within utterances, such as linking the root words, nouns (N), adjectives (ADJ), and verbs (V) between two utterances. However, these approaches resulted in performance degradation. We infer that such heuristic methods significantly disrupt utterance dependencies, leading to an increased introduction of noise and a tendency to focus on irrelevant information. In contrast, the CU-Parser dynamically achieves a balanced fusion of dependency information both within and across utterances.

(2) Both internal and external syntactic associations are crucial. One possible explanation is that these syntactic connections shorten the dependency distance between sentiment elements, as detailed in Appendix F. Further analysis in Appendix F shows that CU-Parser most reduces the dependency distance for the $\mathcal{A}$-$\mathcal{O}$ relationship against $\mathcal{T}$-$\mathcal{A}$ and $\mathcal{T}$-$\mathcal{O}$, leading to the greatest improvement in $\mathcal{A}$-$\mathcal{O}$ matching, supporting our hypothesis.

(3) CU-Parser achieved good performance with both SuPar and Stanford Parser. Overall, SuPar outperformed Stanford Parser in parsing accuracy (Liang et al., 2022), giving CU-Parser a slight performance advantage when equipped with SuPar in DiaASQ, which aligns with the parsers' accuracy.

7 Conclusion

In this study, we propose the MARN for the DiaASQ task. This framework enhances utterance-level associations by leveraging holistic semantics and improves word-level associations through cross-utterance syntax captured by a newly developed CU-Parser. Additionally, MARN incorporates a multi-view data augmentation strategy for low-resource settings. Experiments on benchmark datasets demonstrate that MARN consistently outperforms baselines. We hope our contributions will offer meaningful insights into this field.

Limitations

To fully understand our scheme MARN, we also analyze its limitations. Although MARN achieves state-of-the-art performance in extensive experimental settings, it still leaves potential improvements for the future work:

- **Type of connections obtained by CU-Parser.** We observed that CU-Parser not only provides connections between words but also includes the types of these connections (*e.g.*, nsubj, conj in syntactic dependency trees), which may carry valuable syntactic information. Therefore, it is worth exploring the potential contribution of these connection types to the DiaASQ task, particularly for cross-utterance word-pair relationships. We plan to investigate this further in our future work.
- **Improved utterance concatenation.** In modeling cross-utterance syntactic dependencies at the word level, we heuristically concatenate two utterances using the conjunction "and". While this serves as a practical approach, it is worth investigating more refined methods for sentence concatenation. We leave it as our future work to explore more seamless and contextually appropriate concatenation strategies to yield more reliable results.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.62276110, No.62172039, and in part by the fund of the Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank reviewers for their comments on improving the quality of this paper.

References

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350.

Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022a. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2974–2985.

Yuqi Chen, Chen Keming, Xian Sun, and Zequn Zhang. 2022b. A span-level bidirectional network for aspect sentiment triplet extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4300–4309.

Zhexue Chen, Hong Huang, Bang Liu, Xuanhua Shi, and Hai Jin. 2021. Semantic and syntactic enhanced aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1474–1483.

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 49–54.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020*, pages 2006–2013. IOS Press.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. Mvp: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4380–4397.

Yingjie Gu, Xiaoye Qu, Zhefeng Wang, Yi Zheng, Baoxing Huai, and Nicholas Jing Yuan. 2022. Delving deep into regularity: a simple but effective method

for chinese named entity recognition. *arXiv preprint arXiv:2204.05544*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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.

Bobo Li, Hao Fei, Fei Li, Yuhan Wu, Jinsong Zhang, Shengqiong Wu, Jingye Li, Yijiang Liu, Lizi Liao, Tat-Seng Chua, et al. 2023a. Diaasq: A benchmark of conversational aspect-based sentiment quadruple analysis. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Bobo Li, Hao Fei, Lizi Liao, Yu Zhao, Fangfang Su, Fei Li, and Donghong Ji. 2024a. Harnessing holistic discourse features and triadic interaction for sentiment quadruple extraction in dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18462–18470.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuan-Jing Huang, and Xipeng Qiu. 2023b. Codeie: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353.

Yuqing Li, Wenyuan Zhang, Binbin Li, Siyu Jia, Zisen Qi, and Xingbang Tan. 2024b. Dynamic multi-scale context aggregation for conversational aspect-based sentiment quadruple analysis. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11241–11245. IEEE.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024c. Knowcoder: Coding structured knowledge into llms for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8758–8779. Association for Computational Linguistics.

Shuo Liang, Wei Wei, Xian-Ling Mao, Fei Wang, and Zhiyong He. 2022. Bisyn-gat+: Bi-syntax aware graph attention network for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1835–1848.

Cheng Liu, Wei Xiang, and Bang Wang. 2024. Identifying while learning for document event causality identification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3815–3827. Association for Computational Linguistics.

Yifan Liu, Wei Wei, Jiayi Liu, Xianling Mao, Rui Fang, and Dangyang Chen. 2022. Improving personality consistency in conversation by persona extending. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1350–1359.

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. *CoRR*, abs/1907.11692.

Zhenyi Lu, Jie Tian, Wei Wei, Xiaoye Qu, Yu Cheng, Wenfeng Xie, and Dangyang Chen. 2024. Mitigating boundary ambiguity and inherent bias for text classification in the era of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7841–7864, Bangkok, Thailand. Association for Computational Linguistics.

Zhenyi Lu, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Dangyang Chen, and Jixiong Chen. 2023. Miracle: Towards personalized dialogue generation with latent-space multiple personal attribute control. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5933–5957.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3538–3547.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13543–13551.

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. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. 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 5679–5688.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021a. Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2910–2922.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021b. Enhancing aspect-level sentiment analysis with word dependencies. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 3726–3739.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021c. Dependency-driven relation extraction with attentive graph convolutional networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4458–4471.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving relation extraction through syntax-induced pre-training with dependency masking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1875–1886.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Introducing syntactic structures into target opinion word extraction with deep learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8947–8956.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9122–9129.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, and Yi Chang. 2021. Eliminating sentiment bias for aspect-level sentiment classification with unsupervised opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3002–3012.

Ziyang Wang, Wei Wei, Shanshan Feng, Xian-Ling Mao, Minghui Qiu, Dangyang Chen, and Rui Fang. 2023. Exploiting group-level behavior pattern for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, and Yuchong Hu. 2019. Emotion-aware chat machine: Automatic emotional response generation for human-like emotional interaction. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1401–1410.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585.

Lulu Xiao, Xiaoye Qu, Ruixuan Li, Jun Wang, Pan Zhou, and Yuhua Li. 2020. Fine-grained text sentiment transfer via dependency parsing. In *ECAI 2020*, pages 2228–2235. IOS Press.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 583–593.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4755–4766.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

Yunyi Yang, Kun Li, Xiaojun Quan, Weizhou Shen, and Qinliang Su. 2020. Constituency lattice encoding for aspect term extraction. In *Proceedings of the 28th international conference on computational linguistics*, pages 844–855.

Chen Zhang, Qiuchi Li, Dawei Song, and Benyou Wang. 2020a. A multi-task learning framework for opinion triplet extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 819–828.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219.

Yu Zhang, Zhenghua Li, and Zhang Min. 2020b. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of ACL*, pages 3295–3305.

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. 2024. Linkner: Linking local named entity recognition models to large language models using uncertainty. In *Proceedings of the ACM on Web Conference 2024*, pages 4047–4058.

Sen Zhao, Wei Wei, Xian-Ling Mao, Shuai Zhu, Minghui Yang, Zujie Wen, Dangyang Chen, and Feida Zhu. 2023. Multi-view hypergraph contrastive policy learning for conversational recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 654–664.

Changzhi Zhou, Zhijing Wu, Dandan Song, Linmei Hu, Yuhang Tian, and Jing Xu. 2024. Span-pair interaction and tagging for dialogue-level aspect-based sentiment quadruple analysis. In *Proceedings of the ACM on Web Conference 2024*, pages 3995–4005.

Yuxiang Zhou, Lejian Liao, Yang Gao, Zhanming Jie, and Wei Lu. 2021. To be closer: Learning to link up aspects with opinions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3899–3909.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Min Zhang. 2021. Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. *arXiv preprint arXiv:2112.06013*.

Tong Zhu, Junfei Ren, Zijian Yu, Mengsong Wu, Guoliang Zhang, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, and Min Zhang. 2023. Mirror: A universal framework for various information extraction tasks. *arXiv preprint arXiv:2311.05419*.

Ding Zou, Wei Wei, Feida Zhu, Chuanyu Xu, Tao Zhang, and Chengfu Huo. 2024. Knowledge enhanced multi-intent transformer network for recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1–9.

A Dataset and Implementation Detail

A.1 Dataset Statistic

We evaluate our models on both English and Chinese datasets proposed by (Li et al., 2023a), which mainly focus on the mobile phone domain. Dataset statistics are shown in Table 5.

A.2 Implement Details of MARN

We conduct experiments on a Nvidia GeForce 4090 GPU, with CUDA 12.1 and PyTorch 1.13.1. The model yielding the optimal performance on the validation set is selected for testing. The total number of parameters of PLMs used in Chinese and English datasets is about 110M and 350M, respectively. We set the temperature to 0 for all LLMs. The number of layers in the RGCN of the DOM and the MGAT of the Syntax and Semantic-based Aggregation are set to 3 and 2, respectively. Batch size is set to 2. AdamW optimizer is adopted with a linear warm-up for the first 10% of steps. The learning rate is configured as 2e-5 for the PLM and 1e-4 for the other modules. The function f represents a two-layer fully connected network utilizing LeakyReLU as the activation function, with a dropout rate 0.4. To control the balance of various loss, γ, β and η are set to 1, 3, and 5, respectively. The optimization confidence weight λ is set to 0.7, with the grid search process and analysis for this hyperparameter detailed in Section A.4.

A.3 Implement Details of Baselines

We use the same backbone to compare our MARN framework with other baselines, including ABSA baselines (EC-ACOS, SpERT, Span-ASTE) and DiaASQ baselines (Meta-WP, H2DT, STS, DMCA). Specifically, RoBERTa-Large is used for the English datasets, while Chinese-RoBERTa-wwm-ext is employed for the Chinese datasets. For the generative ParaPhrase baseline, we use mT5-base (Xue et al., 2021) as its backbone, following the configurations outlined in other previous works (Li et al., 2023a, 2024a; Zhou et al., 2024) to ensure a fair comparison. All other hyperparameter settings are consistent with the optimal configuration provided in the original papers.

A.4 Parameter Study for λ

Figure 5 presents the results of varying the optimization confidence threshold. As expected, setting the confidence too low or too high leads to performance degradation. When the confidence is too

Method	English		Chinese	
	S (Dialogue/s)	F1 (%)	S (Dialogue/s)	F1 (%)
MARN	6.02	44.45	8.63	44.72
Vanilla	6.14	39.79	8.85	39.38
STS	5.76	43.80	8.32	40.59

Table 4: "P" represents the total number of parameters, while "S" denotes the training speed, measured in 'Dia/s', which indicates the number of dialogues processed per second, and "F1" indicates the model's performance on the task of sentiment quadruple extraction ($\mathcal{Q}$).

low, the model essentially relies on the traditional DSF-provided utterance associations for sentiment quadruple extraction, failing to address structural irrelevant and implicit associations, as discussed in the main body of this paper. Conversely, setting the confidence too high results in the underutilizing of the valid associations provided by DSF. The experimental results suggest that a threshold value 0.7 strikes an optimal balance.

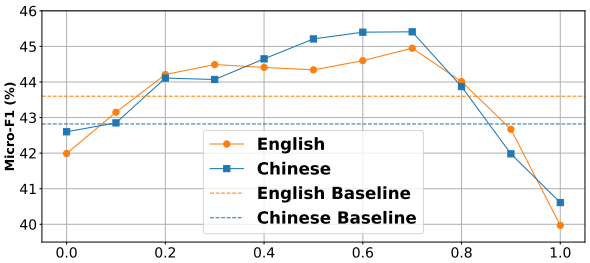


Figure 5: F1 scores on English and Chinese datasets when using different optimization confidence weights. The dotted lines represent the performance of the strongest baseline on both datasets.

B Model Efficiency Analysis

To further analyze the model's efficiency, we provide a detailed comparison of the training time and final performance. As shown in Table 4, we compare MARN with a version that excludes the utterance and syntax-based word association components (Vanilla), as well as with the relatively strongest baseline STS. Compared to the vanilla method, the incorporation of the associate refinement component in MARN results in only a slight increase in training time ($\leq 3\%$). However, MARN achieves a significant performance improvement (from 39% to 44%). We believe this represents an acceptable trade-off between model efficiency and performance gain.

Dataset		#Dia.	#Utt.	#Spk.	#Q.	#IQ.	#CQ.
English	train	800	5947	3897	4414	3442	972
	dev	100	748	502	555	423	132
	test	100	757	503	545	422	123
Chinese	train	800	5974	3987	4607	3549	1013
	dev	100	748	502	577	440	137
	test	100	757	503	558	433	125

Table 5: "#Dia.", "#Utt.", and "#Spk." denote the total number of dialogues, the count of utterances, and the total number of speakers within the corpus, respectively. "#Q." represents the total number of quadruples, while "#IQ." and "#CQ." indicate the number of quadruples contained within the same utterance and those spanning across different utterances, respectively.

C Cross Utterance Relations of Sentiment Quadruples

We also examined the distribution of various types of cross-discourse relations within the three categories of cross-utterance sentiment element pairs. As illustrated in Table 6, a significant majority of the sentiment element pairs involve utterances that belong to a reply relationship, representing about 75% of all cases, with a notable proportion also originating from the same speaker. This underscores the importance of effectively incorporating both relationships into the model, as they offer valuable utterance-level association information.

However, a significant portion of sentiment element relationships do not fall under either of the aforementioned dialogue structural features ("#Other." in Table 6). In other words, it is challenging to construct utterance-level relations that comprehensively capture all sentiment element connections solely based on prior DSF. Since the joint extraction of the sentiment quadruple requires accurate prediction of the association between any two sentiment elements, leveraging the holistic semantic information of utterances to refine the original DSF is crucial, as it enables the discovery of relationships beyond predefined rules.

D Quadruple Decoding Algorithm

This section provides a detailed explanation of the quadruple decoding process, as outlined in Algorithm 1. For the three sentiment elements extracted in subtask 1, candidate triplets (t, a, o) are generated through three traversals. Subsequently, triples are filtered based on whether their internal relationship labels satisfy the matching conditions. Sentiment polarity is then assigned according to the labels from subtask 3 to form valid quadruples. Fig-

(a) Element Extraction (b) Relation Matching (c) Polarity Classification

Figure 6: Decoding example for the sentiment quadruple (*iPhone 14*, *camera quality*, *top notch*, *positive*)

Dataset		$\mathcal{T}$-$\mathcal{A}$			$\mathcal{T}$-$\mathcal{O}$			$\mathcal{A}$-$\mathcal{O}$		
		#Rpy.	#Spk.	#Other.	#Rpy.	#Spk.	#Other.	#Rpy.	#Spk.	#Other.
English	train	72.3%	13.9%	13.8%	71.6%	14.1%	14.3%	85.2%	8.2%	6.6%
	dev	75.2%	15.4%	9.4%	73.2%	18.1%	8.7%	71.4%	17.6%	11.0%
	test	73.6%	12.5%	13.9%	77.8%	10.4%	11.8%	90.2%	2.1%	7.7%
Chinese	train	72.6%	13.9%	13.5%	71.6%	14.0%	14.4%	85.5%	8.4%	6.1%
	dev	75.3%	15.3%	9.4%	74.2%	17.4%	8.4%	71.4%	17.6%	11.0%
	test	74.4%	12.4%	13.2%	77.6%	10.2%	12.2%	92.3%	2.0%	7.7%

Table 6: The proportion of different types of cross-utterance relations for sentiment element pairs. "#Rpy." and "#Spk." respectively represent that the two utterances corresponding to the two sentiment elements (*i.e.*,$\mathcal{T}$-$\mathcal{A}$, $\mathcal{T}$-$\mathcal{O}$, $\mathcal{A}$-$\mathcal{O}$) belong to a reply relationship or belong to the same speaker. "#Other." represents other types of relationships.

Method	English			Chinese		
	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$
MARN	**55.17**	**55.82**	**58.72**	**58.02**	**53.55**	**55.24**
- w/o CU-Parser	53.96	53.83	53.11	57.12	51.60	52.33

Table 7: Ablation results in cross-utterance linguistic features in relation matching subtasks. The values showing the most significant decline in performance across the three relations ($\mathcal{T}$-$\mathcal{A}$, $\mathcal{T}$-$\mathcal{O}$, $\mathcal{A}$-$\mathcal{O}$) are highlighted with an underscore.

Dataset	Method	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$
English	CU-Parser	55.17	55.82	58.72	48.89	44.95
	- w SSA	55.34	56.13	58.56	49.12	45.09
	- w LLM	**55.78**	**56.22**	**59.44**	**49.65**	**45.42**
Chinese	CU-Parser	58.02	**53.55**	55.24	48.23	45.41
	- w SSA	58.31	53.19	55.23	48.30	45.46
	- w LLM	**58.66**	53.17	**55.94**	**48.87**	**46.04**

Table 8: Experiment results of other semantic-based connection methods used in CU-Parser.

ure 6 shows a decoding example for a quadruple.

E Further Exploration of CU-Parser

E.1 Semantic-based Method

In Section 4.3, we introduced the CU-Parser, which connects two utterances using "and" to further construct cross-utterance syntactic relations. Since "and" is typically used to express coordination or progression, it largely preserves the semantic coherence and logical relationships between the sentences when their topics are consistent, thereby facilitating overall parsing. In addition, we further explored two more complex, semantics-based connection methods for utterances: one based on **Sentence-Level Sentiment Analysis (SSA)** and the other based on **LLMs**:

- **CU-Parser w SSA**. To ensure that conjunctions align with the overall emotional transitions between utterances, we perform cross-utterance connections based on their sentiment orientations. Since SSA is a well-established task, we utilize pre-trained open-source models [2] for SSA. Specifically, we connect two utterances with "and" when they share the same sentiment (e.g., positive-positive), with "but" when they have opposing sentiments (e.g., positive-negative), and concatenate them directly if both are neutral.
- **CU-Parser w LLM**. We directly use LLM (GPT-3.5-turbo) to select appropriate conjunctions for splicing based on the overall semantics of the two utterances.

The experimental results are shown in Table 8. It is evident that using SSA or LLMs to select conjunctions based on the overall semantic

[2] https://huggingface.co/tabularisai/multilingual-sentiment-analysis

Algorithm 1 Decoding Algorithm for MARN

Input: Extraction results $tag^{\mathcal{C}}_{ab,cd}$ of word pairs for the three subtasks, where $\mathcal{C} \in \{ele, rel, pol\}$.
Output: Targets $\mathcal{T}$, Aspects $\mathcal{A}$, Opinions $\mathcal{O}$, Sentiment Triplets $\mathcal{S}$ and Quadruples $\mathcal{Q}$.

Initialize the target set $\mathcal{T}$, aspect set $\mathcal{A}$, opinion set $\mathcal{O}$, entity relation set $\mathcal{R}$, and quadruple set $\mathcal{Q}$ with $\varnothing$.
$\mathcal{T} = \{(ab, ac) | tag^{ele}_{ab,ac} = T, b \leq c\}$
$\mathcal{A} = \{(ab, ac) | tag^{rel}_{ab,ac} = A, b \leq c\}$
$\mathcal{O} = \{(ab, ac) | tag^{pol}_{ab,ac} = O, b \leq c\}$
$\mathcal{R}_{TA}, \mathcal{R}_{TO}, \mathcal{R}_{AO} =$ Head-Tail$(\mathcal{T}, \mathcal{A}, \mathcal{O}, tag^{rel})$
while $t \in \mathcal{T}$, $a \in \mathcal{A}$ and $o \in \mathcal{O}$ **do**
  $\mathcal{S} \leftarrow \varnothing$
  if $Tuple(t, a) \in \mathcal{R}_{TA}$ **then**
    if $Tuple(t, o) \in \mathcal{R}_{TO}$ **then**
      if $Tuple(a, o) \in \mathcal{R}_{AO}$ **then**
        Senti $= \{tag^{pol}_{t,a}, tag^{pol}_{t,o}, tag^{pol}_{a,o}\}$. The sentiment label with the highest count will be denoted as s.
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(t, a, o)\}$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(t, a, o, s)\}$
      end if
    end if
  end if
end while
return the set $\mathcal{T}$,$\mathcal{A}$,$\mathcal{O}$,$\mathcal{S}$,$\mathcal{Q}$

Dataset		English			Chinese		
		train	valid	test	train	valid	test
h2h	$\mathcal{T}$-$\mathcal{A}$	2.85	2.74	2.89	3.51	3.67	3.61
	$\mathcal{T}$-$\mathcal{O}$	2.75	2.75	2.71	3.43	3.59	3.45
	$\mathcal{A}$-$\mathcal{O}$	**2.20**	**2.13**	**2.06**	**2.12**	**2.25**	**2.04**
t2t	$\mathcal{T}$-$\mathcal{A}$	3.19	3.22	3.21	3.57	3.75	3.62
	$\mathcal{T}$-$\mathcal{O}$	3.06	2.98	3.06	3.77	3.85	3.70
	$\mathcal{A}$-$\mathcal{O}$	**2.68**	**2.60**	**2.67**	**3.08**	**3.07**	**3.07**

Table 9: The average dependency distances of the three sentiment elements. The terms "h2h" and "t2t" denote the head-head and tail-tail word connections between sentiment elements, respectively. The numbers in bold represent the shortest distance in $\mathcal{T}$-$\mathcal{A}$, $\mathcal{T}$-$\mathcal{O}$, and $\mathcal{A}$-$\mathcal{O}$.

orientation of utterance does result in a slight performance improvement, though the effect is not significant (< 0.8%). It is expected, as the primary task of a dependency parser is to establish syntactic relationships between words, focusing more on the grammatical connections between terms rather than the deeper semantic structures between sentences. Therefore, even with different conjunctions, the parser may not differentiate significantly when processing cross-utterance dependencies. Furthermore, these methods inevitably increase the time required for data preprocessing or incur additional API calls. Hence, we conclude that using "and" for heuristic cross-utterance connections strikes a balance between effectiveness and convenience.

E.2 Detailed Analysis

Traditional approaches to modeling syntactic dependencies are confined to a single sentence. In contrast, our method unifies the CU-Parser and DOM to achieve **joint** and **discriminative** modeling of word associations across multi-utterance. This synergy enables two key capabilities:

- **Joint Association Modeling**. We extend conventional sentence-level syntax parsing to capture cross-utterance links. A unified concatenation mechanism (Section 4.3) allows word associations to span utterance boundaries.
- **Discriminative Association Calibration**. The DOM provides utterance-level association signals, which CU-Parser leverages to dynamically adjust the salience of cross-utterance word associations. Strong utterance-level associations trigger amplified word dependency connections, while weak associations suppress irrelevant dependency links, ensuring context-aware prioritization.

To validate the effectiveness of these capabilities for DiaASQ, we conducted the experiment in Table 10. The results show a significant drop in performance when CU-Parser is removed, confirming the effectiveness, as discussed in Section 5.4. Similarly, removing the utterance association information from DOM also reduces performance, highlighting the importance of the discriminative association calibration achieved by integrating CU-Parser with DOM. Thus, combining utterance-level associations maximizes CU-Parser's potential.

In summary, the **core innovation** lies in this dual mechanism: CU-Parser establishes a holistic word-level syntactic link graph, while discriminative refines it via DOM-guided attention. Together, they transcend utterance-isolated analysis, enabling cross-utterance word association that adaptively emphasizes semantic and syntax relationships.

F Word Pair Dependency Distance

Table 7 reports the results of an ablation study evaluating the impact of syntactic features provided by CU-Parser on the sentiment element relationship matching subtask. Observations are:

(1) Removing syntactic features results in a noticeable decline in model performance for relationship matching across both datasets, underscoring their critical role, which enhances the mutual perception of sentiment element pairs within and across utterances.

Model	English		Chinese	
	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$
MARN	**48.89**	**44.95**	**48.23**	**45.41**
- w/o DOM in CU-Parser	46.99	42.35	47.26	43.74
- w/o CU-Parser	45.01	41.12	44.25	41.70

Table 10: Ablation results. "w/o CU-Parser" refers to completely removing syntax-based MGAT, "w/o DOM in CU-Parser" refers to the removal of the utterance-level association information provided by DOM in the syntax-based MGAT, while maintaining it unchanged in the semantic-based MGAT.

(2) The most significant performance drop occurs in the matching of Aspect-Opinion pairs. We attribute this to the dependency distance, defined as the number of connection edges between two words in the syntactic dependency graph generated by CU-Parser. As shown in Table 9, the average dependency distance between aspect-opinion pairs is shorter than that of other sentiment element pairs in both the Chinese and English datasets, suggesting that syntactic features are particularly beneficial when dependency distances are shorter.

Moreover, we have observed that in many cases, a target (e.g., a product) and its corresponding opinion or aspect may not appear within the same syntactic unit. To address this, we propose designing a dedicated parser for syntactic relationships across utterances to enhance the modeling of sentiment element connections across sentences, which we plan to explore in future work. Additionally, we aim to investigate techniques beyond syntactic parsing to identify when different parts of a sentence or multiple sentences refer to the same entity. This could improve the linking of targets with related opinions or aspects, complementing syntactic methods and offering a more comprehensive approach to extracting sentiment element pairs.

G Experiments with Other LLMs

G.1 Implement Detail and Results

To comprehensively evaluate the performance of LLMs on the DiaASQ task, we conducted experimental comparisons between MARN and various families of LLMs (LLaMa[3], Qwen[4], and Mistral[5]). Considering the substantial overhead of supervised fine-tuning (SFT) (Lu et al., 2024) and to ensure a fairer comparison, we compared MARN (350M and 110M for English and Chinese datasets respectively) with the LoRA (Hu et al., 2021) fine-tuning performance of ~7B-scale LLMs and the in-context learning (ICL) performance of ~70B-scale LLMs. Appendix J details the prompt used for LoRA and ICL. Table 12 gives experiment results. We can observe that even when leveraging the full dataset to perform LoRA fine-tuning on ~7B-scale LLMs or utilizing larger ~70B-scale LLMs, these models still face challenges compared to MARN.

[3] https://huggingface.co/meta-llama
[4] https://huggingface.co/Qwen
[5] https://huggingface.co/mistralai

G.2 Analysis and Further Improvement

As several existing studies have shown (Li et al., 2024c; Zhou et al., 2024), LLMs still exhibit limited performance on complex information extraction tasks similar to the DiaASQ task. We hypothesize that the suboptimal performance of LLMs may be attributed to two factors: **the lack of intuitiveness in describing DSF** and **the complexity of the output format**. As shown in Table 15, the prompts directly input the response and speaker information from the DSF in a list format, which may hinder the model's ability to accurately capture the associations between utterances, resulting in poor performance. In addition, the output contains multiple metrics, which may make the task too complex and prevent the model from focusing solely on extracting sentiment quadruples. Therefore, we tried the following three methods to optimize the prompts used when applying LLM:

- **Prompt Optimization (PO)**. To make DSF more intuitive, we optimize the prompts by marking speaker information directly before each utterance and appending the corresponding reply information at the end of each utterance, as shown in Table 16.
- **Only Quadruple Extraction (OQE)**. We simplify the extraction task by focusing exclusively on extracting the sentiment quadruple.
- **Code-style Prompt (Code)**. Inspired by previous works (Li et al., 2023b; Sainz et al., 2024), we use code-style prompts to standardize structured output and deploy the corresponding code version of LLM.

The experimental results indicate that the optimized prompts improve performance to some extent. Furthermore, directly outputting the quadruple extraction results and using code-style prompts also yield notable improvements. Therefore, enhancing LLM's understanding of dialogue structural fea-

Method	Model	English								Chinese							
		$\mathcal{T}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$	$\mathcal{T}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$
-	**MARN**	**89.53**	**76.22**	**61.75**	**55.17**	**55.82**	**57.69**	**48.89**	**44.95**	**92.34**	**77.20**	**61.94**	**58.02**	**53.55**	**55.24**	**48.23**	**45.41**
LoRA	LLaMa3.1-8B	74.60	51.35	44.03	43.23	40.25	35.91	27.93	25.36	77.46	68.28	51.57	41.09	34.92	31.03	28.05	27.49
	Qwen2-7B	73.48	61.75	46.28	48.33	44.27	41.02	32.19	31.36	81.34	69.39	53.49	42.39	36.38	34.27	30.72	28.36
	Mistral-7B	79.34	66.62	49.23	46.12	42.97	39.65	33.18	30.38	80.35	70.31	48.29	42.10	35.27	33.92	28.10	26.58
ICL	LLaMa3.1-70B	62.58	53.56	57.14	39.27	31.05	28.31	23.48	21.92	45.18	44.25	49.29	41.38	31.48	26.79	23.71	19.35
	Qwen2-72B	63.46	41.36	51.26	29.93	26.09	31.29	22.76	22.36	50.98	48.55	52.05	50.55	30.32	27.71	25.37	22.11
	Mistral-Large	60.33	42.11	52.93	33.47	31.84	25.31	22.34	19.59	49.27	49.02	49.34	47.28	34.29	25.38	22.93	20.49

Table 11: We compare our MARN against LLaMa, Qwen, and Mistral series LLMs, employing a 3-shot approach for few-shot demonstrations.

Method	Strategy	English								Chinese							
		$\mathcal{T}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$	$\mathcal{T}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$	$\mathcal{T}$-$\mathcal{O}$	$\mathcal{A}$-$\mathcal{O}$	$\mathcal{T}$-$\mathcal{A}$-$\mathcal{O}$	$\mathcal{Q}$
-	**MARN**	**89.53**	**76.22**	**61.75**	**55.17**	**55.82**	**57.69**	**48.89**	**44.95**	**92.34**	**77.20**	**61.94**	**58.02**	**53.55**	**55.24**	**48.23**	**45.41**
LoRA	Vanilla	73.48	61.75	46.28	48.33	44.27	41.02	32.19	31.36	81.34	69.39	53.49	42.39	36.38	34.27	30.72	28.36
	PO	74.02	61.23	45.09	51.31	46.23	45.15	34.14	32.18	82.17	70.38	52.62	46.23	40.55	38.21	35.12	32.27
	OQE	/	/	/	/	/	/	/	32.89	/	/	/	/	/	/	/	32.65
	Code	80.24	64.12	48.28	49.12	45.78	41.35	33.47	32.00	82.11	69.34	54.92	43.35	37.23	36.23	31.56	29.67
	PO+OQE	/	/	/	/	/	/	/	34.62	/	/	/	/	/	/	/	35.72
ICL	Vanilla	63.46	41.36	51.26	29.93	26.09	31.29	22.76	22.36	50.98	48.55	52.05	50.55	30.32	27.71	25.37	22.11
	PO	65.74	43.21	55.39	34.34	32.35	35.04	27.45	24.10	49.23	51.09	53.46	52.14	34.92	31.39	28.35	26.05
	OQE	/	/	/	/	/	/	/	24.35	/	/	/	/	/	/	/	24.01
	Code	64.66	43.38	54.38	32.10	30.33	34.36	25.99	23.93	49.39	50.11	52.59	51.63	33.29	30.81	26.35	24.29
	PO+OQE	/	/	/	/	/	/	/	26.39	/	/	/	/	/	/	/	26.94

Table 12: We compare the performance of MARN with the results of LLMs employing different optimization strategies. Specifically, the LLMs utilizing the "PO" and "OQE" strategies are based on Qwen2-7B-Instruct, while the LLM employing the "Code" strategy is based on Qwen2.5-Coder-7B-Instruct.

tures or refining the output format emerges as a promising research direction, which will also be the focus of our future work.

H Details about Multi-level Data Augmentation

H.1 Implement Detail

In our experiments, the LLM used for data augmentation was ChatGPT. The total number of augmented instances, k, was set to 6, indicating that both word-level and utterance-level augmentation underwent 3 rounds each. For each round, 50% of the utterances were randomly selected for augmentation, and all the scores are averaged values over five runs under different random seeds.

H.2 Prompt for Data Augmentation

Prompt for Word-level Augentation

```
Please replace the key backbone
words in the sentence with their
synonyms while keeping the overall
meaning and the other words
unchanged:
{Utterance}
```

Prompt for Utterance-level Augentation

```
Please help me rephrase the
following sentence while preserving
its original meaning:
{Utterance}
```

H.3 Detailed Analysis

This section conducts a more detailed analysis to validate the efficacy of our data augmentation strategy and explore the underlying reasons for its effectiveness. Specifically, it includes an analysis of the impact of the JS divergence in utterance associations, as well as the influence of data distribution and its differences on the results.

H.3.1 The Impact of JS Divergence

Data augmentation primarily strengthens the associations between utterances, and we use JS divergence to quantify the difference in utterance associations between the augmented data and the corresponding original data. A larger JS divergence indicates that the model finds it more challenging to align the utterance associations of the augmented data with those of the original data in similar contexts, signaling computational instability at

Dataset	Augment Data	English (Batch of Train)			Chinese (Batch of Train)			English (Test)			Chinese (Test)		
		BoW	TF-IDF	Cos_Sim	BoW	TF-IDF	Cos_Sim	BoW	TF-IDF	Cos_Sim	BOW	TF-IDF	Cos_Sim
Batch 1	1st Performance	3rd	2nd	3rd	3rd	3rd	3rd	2nd	3rd	2nd	1st	2nd	2nd
	2nd Performance	2nd	3rd	2nd	2nd	2nd	2nd	3rd	1st	3rd	3rd	3rd	3rd
	3rd Performance	1st	1st	1st	1st	1st	1st	1st	2nd	1st	2nd	1st	1st
Batch 2	1st Performance	3rd	3rd	2nd	3rd	3rd	3rd	2nd	3rd	3rd	2nd	3rd	2nd
	2nd Performance	2nd	2nd	1st	2nd	2nd	2nd	1st	1st	2nd	1st	2nd	1st
	3rd Performance	1st	1st	3rd	1st	1st	1st	3rd	2nd	1st	3rd	1st	3rd
Batch 3	1st Performance	3rd	3rd	3rd	3rd	3rd	3rd	3rd	2nd	1st	2nd	1st	2nd
	2nd Performance	2nd	2nd	2nd	2nd	2nd	2nd	1st	3rd	3rd	3rd	2nd	3rd
	3rd Performance	1st	1st	1st	1st	1st	1st	2nd	1st	2nd	1st	3rd	1st

Table 13: The similarity rank between the augmented data with original data and test set. The "Augment Data" column represents the performance ranking of the model on the quadruple extraction task after training with data augmented three times per batch. The numbers in the table indicate the ranking of the similarity between the augmented data and both the original data and the test data, computed after three augmentation steps for each batch.

the level of utterance associations. Specifically, we conducted the following experiment to analyze the specific impacts of this difference on model performance: We randomly selected 20% of the data from the training set three times to simulate three batch low-resource conditions and trained using this data to obtain preliminary test results. Subsequently, by varying the temperature parameter of the LLMs three times, we generated three sets of augmented data based on the selected data of each batch. Using the trained model, we then calculated the average JS divergence between the training data and the augmented data of the association of utterances. The augmented data was then merged with the original data, and the combined dataset was used for retraining from scratch. Finally, the model's performance trained on the merged dataset was evaluated as shown in the Figure 8. We use Min Div, Mid Div, and Max Div to represent the average JS divergence rank between the utterance association matrices of the augmented data and the original data after each of the three data augmentation processes for each batch. Observations are: (1) Due to multi-granularity data augmentation, the training processes that utilized augmented data consistently demonstrated improved performance. (2) Overall, the greater the difference in JS divergence between the augmented data and the original data's utterance associations, the more significant the improvement (performance increases as the JS divergence increases). We further infer that when the JS divergence between the original and augmented data is larger, the augmented data better simulates a broader range of real-world dialogue scenarios. This compensates for the shift in data distribution and underrepresentation caused by the reduced size of the original data in low-resource settings.

However, it is important to note that, since the prompts used for LLMs during the data augmentation process already constrain the output to follow the original dialogue information, the augmented data does not deviate excessively from the real distribution. In other words, this guides us to ensure that, during the data augmentation process, the strategy should focus on increasing the JS divergence while maintaining the overall semantic consistency of the dialogue, avoiding significant catastrophic deviation from the original data.

H.3.2 Generalization Ability Verification

To demonstrate that data augmentation improves the model's generalization ability rather than merely overfitting the test set, we conducted the experiment outlined in Table 13. Specifically, we employed three metrics: Bag of Words (BoW), TF-IDF, and cosine similarity of word embeddings[6] to calculate the similarity between the augmented data with original and test data. For each augmented sample, we matched it with the most similar instance from the corresponding dataset and computed the average similarity. The results reveal that as the similarity between the augmented data and the original data decreases, the performance improves, which aligns with the conclusion in Appendix H.3.1. More importantly, data that is more similar to the test set does not necessarily lead to better performance. That is, better results can be achieved without the augmented data closely matching the test set. This suggests that the performance improvement due to data augmentation is not because the augmented data is more similar to the test set, but because data augmentation effectively enhances the generalization capability.

[6] https://nlp.stanford.edu/projects/glove/

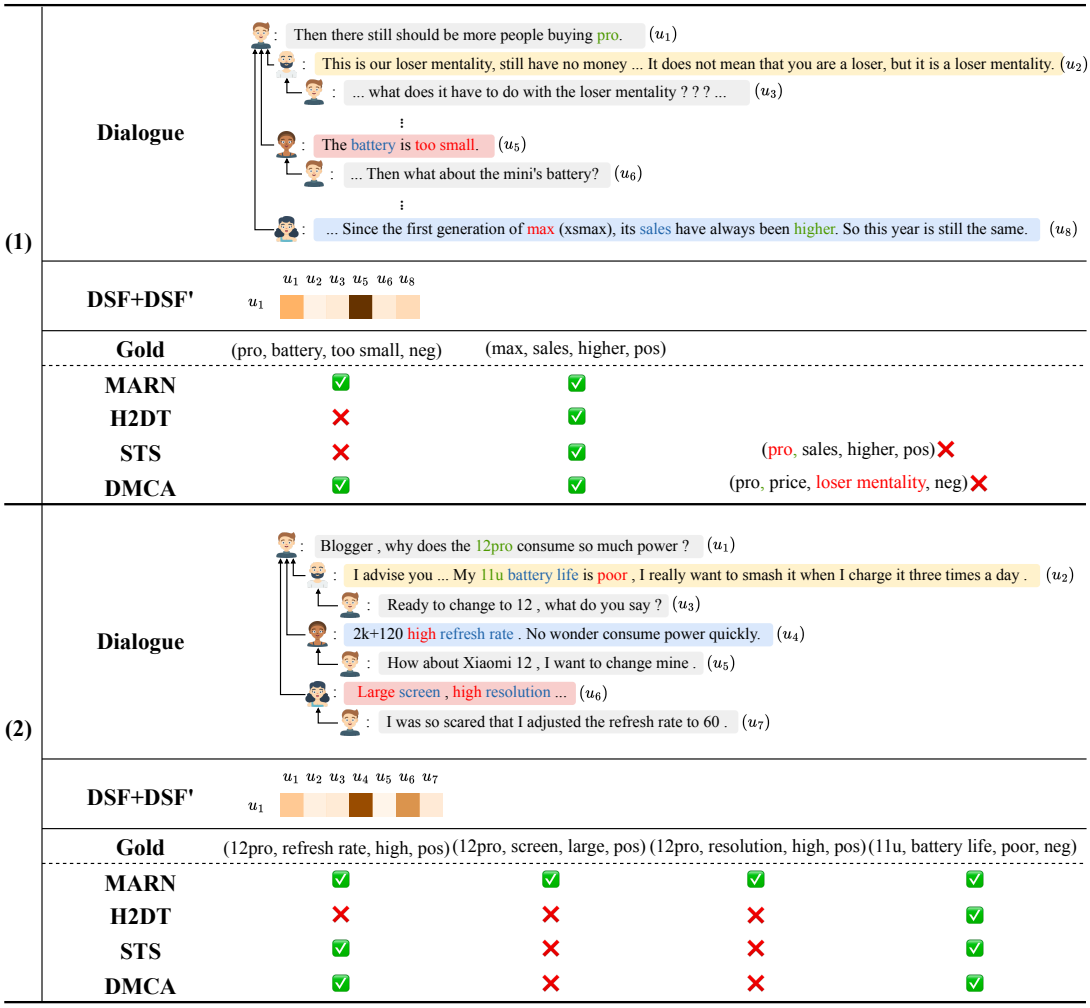


Figure 7: Example cases demonstrating the ground truth labels alongside the predictions generated by three baseline models and our proposed model. Arrows represent reply relationships. Utterances belonging to the same speaker are marked in the same color, such as u_1, u_3, and u_6 in case 1.

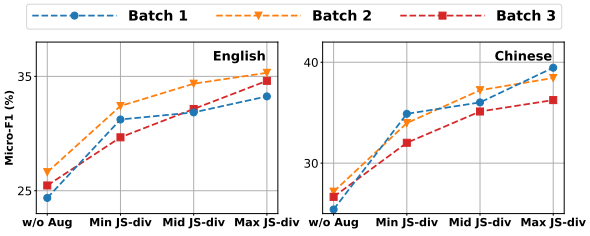


Figure 8: The model performance across four conditions from data augmentation: **w/o Aug** represents the method with no augmentation. **Min Div** refers to the augmentation with the smallest JS-divergence in each batch, indicating minimal change in utterance associations. **Mid Div** corresponds to the augmentation with moderate divergence in each batch, while **Max Div** reflects the largest divergence in each batch, introducing the most significant variation in utterance associations compared to the original.

I Case Study

As shown in Figure 7, we present two examples to help better understand our proposed model. Observations are:

(1) We provide visualization of the association degree (DSF+DSF′) between u_1 and other utterances in two cases. It can be found that utterances with shared sentiment elements are given higher attention weights. For example, in case (1), u_1 and u_5's discussion target is "pro", so they are assigned the highest association value. Although u_2 and u_8 are also replying to u_1, u_2's topic is not discussing the sentiment element related to u_1, and u_8 is discussing another target "max", so they are both assigned lower attention values.

(2) We show the sentiment quadruple extraction cases of MARN and the other three baselines. We present the results of sentiment quadruple extraction for MARN and three baseline models. It is

observed that when all sentiment elements of a quadruple are contained within a single utterance, all models achieve accurate extraction. However, when sentiment elements are dispersed across multiple utterances, only MARN successfully extracts all elements of the quadruple. We infer this is due to the model's ability to handle complex contextual semantics. Specifically, when the context is more intricate, other models may be misled by irrelevant utterances or struggle to focus on sentiment-relevant utterances. For instance, in case (1), the DMCA model erroneously associates the target "pro" from utterance u_1 with an unrelated opinion "loser mentality" presented in utterance u_2. At the same time, the H2DT overlooks sentiment-related terms such as "battery" and "too small" in utterance u_5. Thus, MARN demonstrates a robust ability to recognize and filter out key information even in complex contextual settings.

J Prompt for LLM Baselines

In this section, as shown in the following three tables, we provide the prompts used with LLM in the baseline, which are divided into three parts: zero-shot prompts, few-shot prompts, and optimized prompts that make DSF more intuitive for the DiaASQ task. For the few-shot setting, we randomly select three examples from the training set to illustrate in-context learning. The ICL-based method and the LoRA-based fine-tuning method use the same few-shot prompt. We set the temperature to 0 for all LLMs across different experiments. To assist the model in understanding the task objectives and dialogue structure features, we provide detailed guidelines for each sentiment element and offer an in-depth explanation of the definitions for each feature included in the original data.

ZERO-SHOT

Q: Given a set of sentiment elements: {"Target", "Aspect", "Opinion", "Sentiment"}, where the meaning of each sentiment element is as follows:

(1) Target: The entity or subject being evaluated.

(2) Aspect: A specific feature or aspect of the target.

(3) Opinion: An evaluative or opinionated term related to the target or aspect.

(4) Sentiment: The overall evaluation or sentiment orientation (positive, negative, neutral) expressed.

Your task is to perform sentiment quadruple extraction at the dialogue level. The input is given in JSON format with the following key components:

(1) Sentences: A list of utterances in the dialogue, where each entry corresponds to an individual utterance.

(2) Replies: An array indicating the reply relationships, where each number corresponds to the index of the reply discourse.

(3) Speakers: An array indicating the speaker for each utterance, where each number represents the speaker index.

Please extract the sentiment elements from the conversation, identify pairs of these elements, and construct the sentiment quadruples. The output should be formatted in JSON as follows:

```
{
    "Targets": [List of Targets],
    "Aspects": [List of Aspects],
    "Opinions": [List of Opinions],
    "Target-Aspect": [List of (Target, Aspect)],
    "Aspect-Opinion": [List of (Aspect, Opinion))],
    "Target-Opinion": [List of (Target, Opinion))],
    "Quadruples": [List of (Target, Aspect, Opinion, Sentiment))],
}
```

DIALOGUE:

```
{
    "Sentences": ["What aspects of the configuration of the big detective...",
                  ...,
                  "To beat the Honor 60Pro casually"],
    "Replies": [-1,0,1,2,0,4],
    "Speakers": [0,1,2,3,4,0]
}
```

ANSWER:

Table 14: Zero-shot prompts for DiaASQ.

FEW-SHOT

Q: Given a set of sentiment elements: {"Target", "Aspect", "Opinion", "Sentiment"}, where the meaning of each sentiment element is as follows:

(1) Target: The entity or subject being evaluated.

(2) Aspect: A specific feature or aspect of the target.

(3) Opinion: An evaluative or opinionated term related to the target or aspect.

(4) Sentiment: The overall evaluation or sentiment orientation (positive, negative, neutral) expressed.

Your task is to perform sentiment quadruple extraction at the dialogue level. The input is given in JSON format with the following key components:

(1) Sentences: A list of utterances in the dialogue, where each entry corresponds to an individual utterance.

(2) Replies: An array indicating the reply relationships, where each number corresponds to the index of the reply discourse.

(3) Speakers: An array indicating the speaker for each utterance, where each number represents the speaker index.

Please extract the sentiment elements from the conversation, identify pairs of these elements, and construct the sentiment quadruples. The output should be formatted in JSON as follows:

```
{
    "Targets": [List of Targets],
    "Aspects": [List of Aspects],
    "Opinions": [List of Opinions],
    "Target-Aspect": [List of (Target, Aspect)],
    "Aspect-Opinion": [List of (Aspect, Opinion)],
    "Target-Opinion": [List of (Target, Opinion)],
    "Quadruples": [List of (Target, Aspect, Opinion, Sentiment)],
}
```

DIALOGUE:

```
{
    "Sentences": ["I hope that x80pro will not use Orion anymore",
                  ...,
                  "At present , X80 will only use the brother and Qualcomm"],
    "Replies": [-1,0,1,2,0,0],
    "Speakers": [0,1,0,1,2,3]
}
```

ANSWER:

```
{
    "Targets": ["x80pro", "X70Pro", ... , "X80"],
    "Aspects": ["charging", "price", ... , "signal"],
    "Opinions": ["attracted", "poor", ... , "not satisfied"],
    "Target-Aspect": [("X70Pro", "charging"), ("fruit", "price"), ...],
    "Aspect-Opinion": [("charging", "always been bad"), ...],
    "Target-Opinion": [("X70Pro", "not satisfied"),  ...],
    "Quadruples": [("x-series", "charging", "always been bad", "neg"),...],
}
```

...

DIALOGUE:

```
{
    "Sentences": ["What aspects of the configuration of the big detective...",
                  ...,
                  "To beat the Honor 60Pro casually"],
    "Replies": [-1,0,1,2,0,4],
    "Speakers": [0,1,2,3,4,0]
}
```

ANSWER:

Table 15: Few-shot prompts for DiaASQ.

OPTIMIZED PROMPT FOR DIAASQ

Q: Given a set of sentiment elements: {"Target", "Aspect", "Opinion", "Sentiment"}, where the meaning of each sentiment element is as follows:

(1) Target: The entity or subject being evaluated.

(2) Aspect: A specific feature or aspect of the target.

(3) Opinion: An evaluative or opinionated term related to the target or aspect.

(4) Sentiment: The overall evaluation or sentiment orientation (positive, negative, neutral) expressed.

Your task is to perform sentiment quadruple extraction at the dialogue level. Please extract the sentiment elements from the conversation, identify pairs of these elements, and construct the sentiment quadruples. The output should be formatted in JSON as follows:

```
{
    "Targets": [List of Targets],
    "Aspects": [List of Aspects],
    "Opinions": [List of Opinions],
    "Target-Aspect": [List of (Target, Aspect)],
    "Aspect-Opinion": [List of (Aspect, Opinion)],
    "Target-Opinion": [List of (Target, Opinion)],
    "Quadruples": [List of (Target, Aspect, Opinion, Sentiment)],
}
```

DIALOGUE:

```
{
    ...,
    "Utterance_i": "Speaker j: {Utterance_context} (reply to utterance_k)",
    ...
}
```

ANSWER:

```
{
    "Targets": ["x80pro", "X70Pro", ... , "X80"],
    "Aspects": ["charging", "price", ... , "signal"],
    "Opinions": ["attracted", "poor", ... , "not satisfied"],
    "Target-Aspect": [("X70Pro", "charging"), ("fruit", "price"), ...],
    "Aspect-Opinion": [("charging", "always been bad"), ...],
    "Target-Opinion": [("X70Pro", "not satisfied"),  ...],
    "Quadruples": [("x-series", "charging", "always been bad", "neg"),...],
}
```

...

DIALOGUE:

```
{
    ...,
    "Utterance_i": "Speaker j: {Utterance_context} (reply to utterance_k)",
    ...
}
```

ANSWER:

Table 16: Optimized Prompt for DiaASQ.