A Joint Coreference-Aware Approach to Document-Level Target Sentiment Analysis

Hongjie Cai, Heqing Ma, Jianfei Yu, Rui Xia*
School of Computer Science and Engineering,
Nanjing University of Science and Technology, China
{hjcai, hqma, jfyu, rxia}@njust.edu.cn

Abstract

Most existing work on aspect-based sentiment analysis (ABSA) focuses on the sentence level, while research at the document level has not received enough attention. Compared to sentence-level ABSA, the document-level ABSA is not only more practical but also requires holistic document-level understanding capabilities such as coreference resolution. To investigate the impact of coreference information on document-level ABSA, we conduct a three-stage research for the document-level target sentiment analysis (DTSA) task: 1) exploring the effectiveness of coreference information for the DTSA task; 2) reducing the reliance on manually annotated coreference information; 3) alleviating the evaluation bias caused by missing the coreference information of opinion targets. Specifically, we first manually annotate the coreferential opinion targets and propose a multi-task learning framework to model the DTSA task and the coreference resolution task jointly. Then we annotate the coreference information with ChatGPT for joint training. Finally, to address the issue of missing coreference targets, we modify the metric from strict matching to a loose matching method based on the clusters of targets. The experimental results demonstrate our framework's effectiveness and reflect the feasibility of using ChatGPT-annotated coreferential entities and the applicability of the modified metric. Our source code is publicly released at https://github.com/NUSTM/DTSA-Coref.

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to extract opinion targets from review texts and the sentiment towards each opinion target (Hu and Liu, 2004). For example, in the sentence "The food is delicious.", the opinion target is "food" and its sentiment is positive.

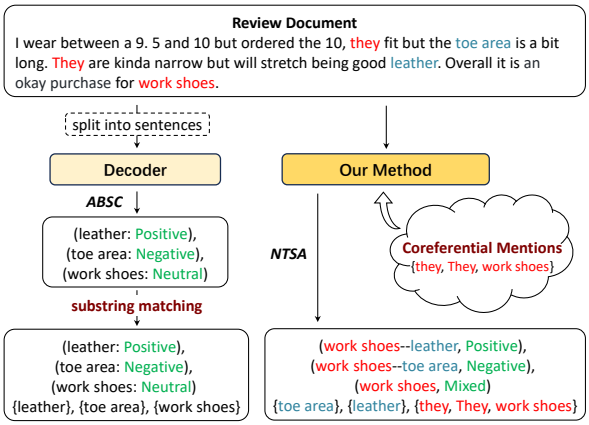

Figure 1: Comparison between previous work and our methodology on document-level target sentiment analysis with coreferential mentions.

Early work (Hu and Liu, 2004; Jiang et al., 2011) employed rules or machine learning methods to solve the task on their annotated datasets. After the introduction of ABSA evaluation tasks and benchmark datasets in (Pontiki et al., 2014, 2015, 2016), subsequent work deepened the research on ABSA, exploring different model paradigms in effects on ABSA (Li et al., 2018; Wu et al., 2020; Zhang et al., 2021a; Gou et al., 2023), also by improving the task definition, providing rich and diverse ABSA datasets to extract opinion targets and their sentiments more accurately (Jiang et al., 2019; Cai et al., 2021; Barnes et al., 2022). Despite these advancements, most existing work focuses on the sentence level, overlooking the fact that product reviews are typically presented in the form of documents, which is not only more practical but also enables the utilization of valuable document-level information for ABSA tasks. Recently, (Luo et al., 2022) introduced a task of document-level target sentiment analysis (DTSA), hierarchically depicting the opinion target. As shown in Figure 1, the review mentions three opinion targets: "work shoes", the "leather" of the work shoes (represented as "work

* Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12149–12160

shoes–leather"), and the "toe area" of the work shoes (represented as "work shoes–toe area"). It expresses a positive sentiment towards "leather", a negative sentiment towards "toe area", and a mixed sentiment towards "work shoes".

A typical phenomenon in the DTSA task is the coreference of opinion targets, especially across multiple sentences. As the example shown in Figure 1, the "work shoes" have two coreferential mentions "they" and "They", and the three aspect terms "work shoes", "they", and "They" refer to the same opinion target. Combining context understanding of these pronouns' referent may help differentiate the sentiment of different opinion targets and accurately identify the hierarchical relation among opinion targets. Although such coreference phenomenon is prevalent in review texts and can affect the accuracy of opinion target extraction in the ABSA research, Very few studies have investigated the coreference problem (Mullick et al., 2023; Rønningstad et al., 2023), especially in the DTSA task, which is relatively complex and has a higher incidence of coreference in review texts. Currently, there is no relevant research addressing coreference problems in this task.

To explore the impact of coreference resolution on the DTSA task, the following three questions should be addressed: First, in the end-to-end DTSA task, does the coreference information contribute to the accurate identification of hierarchical relations among opinion targets and their sentiments? Second, coreference annotation requires substantial human effort and time. Is it possible to mine coreference information without manual efforts? Third, the same opinion target may have different forms of textual expression, yet existing ABSA datasets often select only one expression as the ground truth label. This approach may treat coreferent expressions of the same target as incorrect during evaluation. How can we mitigate the evaluation errors caused by the omission of coreference annotations?

To address the first question, we annotate the coreferential targets of the opinion targets, proposing a coreference-aware joint model to investigate whether the coreference information can improve the model's ability to identify hierarchical opinion targets and their sentiments. The model employs a dual-path architecture, utilizing multi-task learning to jointly handle the DTSA task and coreference resolution task. The left path is designed to identify opinion targets, their sentiments, and the hierarchical relations between them. The right path focuses on identifying the opinion targets and their coreferential relations. By sharing parameters and aligning tokens between the two paths, the model can learn to leverage coreference information and enhance its performance on the DTSA task. In response to the second question, we leverage ChatGPT to annotate the coreference information and verify the efficacy of the coreference annotation in the joint model. To answer the third question, we modify the metric to employ a cluster matching method that takes into account coreferent opinion targets.

The experimental results demonstrate the effectiveness of introducing coreference information through a dual-path model. Additionally, the coreference annotations provided by ChatGPT can achieve comparable DTSA performance to those of human annotations. Finally, the cluster-level evaluation metric is also proven to be more reasonable in assessing the effectiveness of opinion target extraction in consideration of the coreference problem.

2 Related Work

In recent years, ABSA has been extensively studied by researchers, and works are primarily focused on the sentence level, emphasizing either single-element extraction or multi-element extraction.

Sentence-level ABSA. Early sentence-level ABSA (Dong et al., 2014) often constructed datasets sourced from Twitter. Following the introduction of the ABSA task through SemEval evaluations by (Pontiki et al., 2014, 2015, 2016), subsequent research proposed numerous methods for sentence-level ABSC (Ma et al., 2017; Xu et al., 2019; Tang et al., 2020). Additionally, a portion of the work further focused on the end-to-end ABSA task, mainly aiming to jointly extract the aspects and their sentiments within sentences. (Li et al., 2019; Luo et al., 2019; He et al., 2019) have introduced various solutions based on encoder-only models for this task. With the development of generative language models, (Zhang et al., 2021a,b) subsequently proposed solutions based on BART or T5.

Document-level ABSA. A few ABSA work was conducted at the document level. (Hu and Liu, 2004; Ding et al., 2008) annotated document-level ABSA datasets across multiple domains and researched multiple aspect extractions on these datasets. (Chen et al., 2020b) investigated the con-

sistency of the same aspect's sentiments across documents as well as the correlation between sentiments across different aspects. (Luo et al., 2022) introduced the DTSA task and proposed a framework based on BART to solve this task. Furthermore, (Song et al., 2023) introduced an encoder-based Sequence-to-Structure framework to explicitly model the hierarchical relations between entities. Our work, taking DTSA as a starting point, investigates the impact of coreference on DTSA.

Coreference Resolution. Coreference resolution aims to identify all expressions that refer to the same entity from the text. Since (Lee et al., 2017) first introduced a deep learning method for end-to-end coreference resolution, coreference resolution has been increasingly integrated into related downstream tasks. (Hu and Liu, 2004) marked entities requiring pronoun resolution in ABSA datasets. (Ding and Liu, 2010) introduced a coreference classification task for objects and entities in comparative reviews. Building on this, (Chen et al., 2020a) proposed a method for automatic mining and utilizing domain-specific knowledge to address coreferences of entities. Moreover, (Mullick et al., 2023; Rønningstad et al., 2023) investigated whether the coreference resolution of entity is beneficial for the ABSC task.

3 Problem Definition

In traditional sentence-level ABSA tasks, given a review sentence s, the goal is to identify the opinion targets (fine-grained entities or their aspects, collectively referred to as entities) mentioned in the sentence and their corresponding sentiment $\{\ldots,(a_i,p_i),\ldots\}$, where a_i represents the extracted i-th entity, typically presented as a continuous text span in the sentence, and p_i denotes the sentiment towards a_i, such as positive, negative, or neutral.

In the DTSA task, the input extends from a single sentence to a document $\boldsymbol{d} = [s_1,\ldots,s_n]$ consisting of n sentences. The output aims to identify all hierarchical entities and their corresponding sentiments $\{\ldots,(t_i,p_i),\ldots\}$. Unlike the flat entity a_i in sentence-level ABSA, t_i represents a multi-level entity composed of multiple flat entities. We will use hierarchical entities to represent multi-level opinion targets in the following. It can be seen that extracting hierarchical entities in the DTSA task is more challenging than sentence-level ABSA tasks, particularly when the metric requires an exact match with the ground truth. Considering the entity coreference in documents, we also modify the metric from the exact entity-level matching to a cluster-level matching.

Domain	Source	Cluster (#/%)	Entities/Cluster
Book	Humans	519/52.64%	2.91
	ChatGPT	466/47.26%	4.05
Clothing	Humans	526/56.68%	3.23
	ChatGPT	374/40.30%	4.02
Hotel	Humans	458/44.51%	2.38
	ChatGPT	369/35.86%	3.00
Restaurant	Humans	550/58.51%	2.52
	ChatGPT	400/42.55%	3.98

Table 1: Statistics on the annotated coreference clusters between humans and ChatGPT. The third and fourth columns display the number and percentage of documents annotated with coreference clusters, as well as the entities per cluster.

4 Methodology

To investigate whether the information of entity coreference can aid in identifying hierarchical entities and their sentiments, we first annotate the coreferential entities for each entity a_i, forming a coreference cluster $C_i = \{a_{i_0},\ldots,a_{i_k}\}$ (§4.1). Next, we incorporate the coreference information into the DTSA task through multi-task learning, leveraging both human-annotated and machine-annotated coreference information to improve the model's ability to recognize coreference (§4.2).

4.1 Entity Coreference Annotation and Analysis

The coreference of entities is manifested as multiple entities at different positions referring to the same entity. To annotate entity coreference, we need to label these entities at different positions and cluster them into a coreferential entity cluster. We obtain the coreference information of entities through both human annotation and ChatGPT annotation (refer readers to Appendix A for annotation details), respectively.

Statistics and Analysis As shown in Table 1, in each domain, the number of documents with human-annotated coreference is higher than those annotated by ChatGPT. The proportion of human-annotated coreference documents ranges from 44% to 59% of the total documents, while ChatGPT-annotated coreference documents account for 35% to 48%. However, for the documents annotated with coreference, the average number of entities per

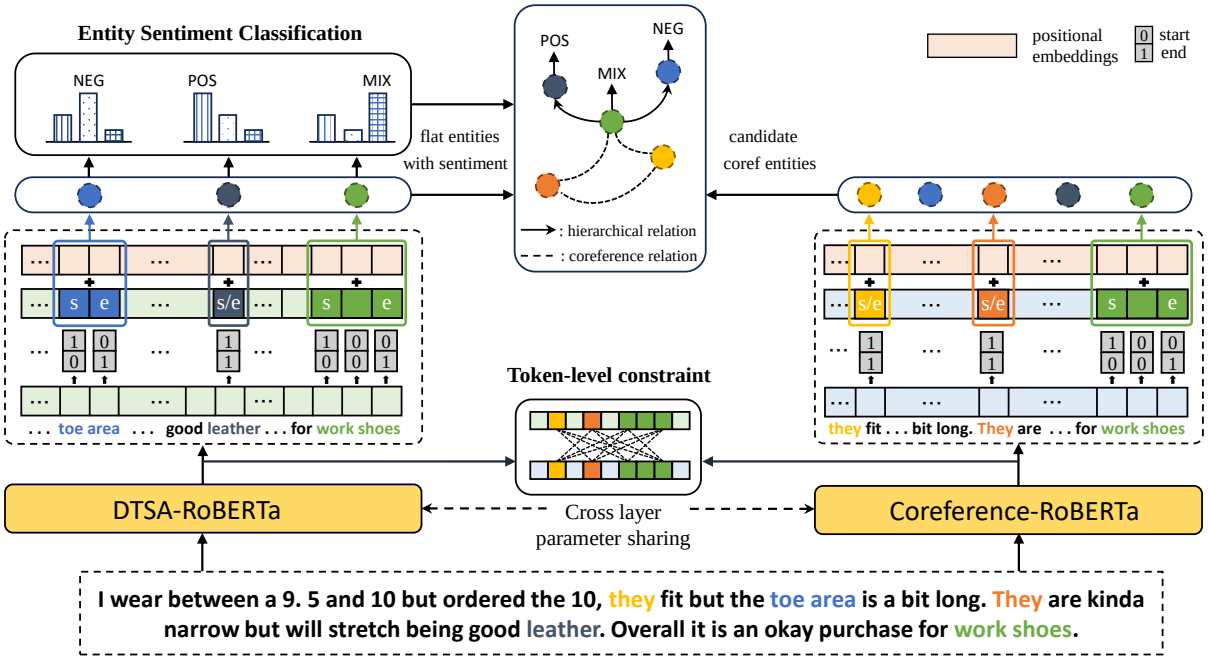

Figure 2: Overview of the joint framework for document-level target sentiment analysis and coreference resolution.

cluster annotated by ChatGPT is generally higher than that annotated by humans, with this number reaching about 4 in the domains of Book, Clothing, and Restaurant, and 3 in the Hotel domain, while the average number of entities per cluster annotated by humans ranges from 2.3 to 3.3. It can be observed that human-annotated coreference covers more examples, but the annotated entities per example are comparatively sparser compared to ChatGPT.

4.2 Joint DTSA and Coreference Resolution Learning

To integrate coreference information into the DTSA task, we employ multi-task learning to model both DTSA and coreference resolution tasks. This allows the model to identify coreference relations between entities and consider the sentiment information surrounding the coreferential entities when determining the sentiment of the current entity.

Specifically, to jointly train the DTSA and coreference resolution tasks, we utilize a dual-path RoBERTa (Liu et al., 2019) to model both tasks simultaneously. As shown in Figure 2, where the two RobBERTa models receive the same input and share parameters.

Backbone We use RoBERTa as the text encoder. Given a document $\boldsymbol{d}$, we can obtain the tokenized tokens $\boldsymbol{X} = \{x_1, \ldots, x_n\}$. After feeding $\boldsymbol{X}$ into RoBERTa, we can get the contextualized representation $\boldsymbol{H}_L = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n\}$ after RoBERTa encoding:

$$\boldsymbol{H}_l = \text{Transformer}_l(\boldsymbol{H}_{l-1}), l \in [1, 12],$$

where $L = 12$ is the number of Transformer layers in RoBERTa-base. Next, we use $\boldsymbol{H}_L$ to extract entities and identify the sentiment of the entities.

4.2.1 DTSA Modeling

We decompose the DTSA task into three subtasks: entity extraction, entity sentiment classification, and entity hierarchical classification.

Entity Extraction Considering that the review texts may contain nested entities (Yan et al., 2022; Wang et al., 2020), and traditional BIO tagging scheme cannot handle these cases, we use a span-based method (Hu et al., 2019) to extract candidate entities. Specifically, we treat the identification of the start and end tokens of an entity as a binary classification task. We perform two binary classifications for each $\boldsymbol{h}_i^m$ in $\boldsymbol{H}_L^m = \{\boldsymbol{h}_1^m, \ldots, \boldsymbol{h}_n^m\}$, determining whether the token is a candidate start token and whether it is a candidate end token, respectively:

$$p(y^s|\boldsymbol{h}_i^m) = \text{Sigmoid}(\boldsymbol{W}_s^m \boldsymbol{h}_i^m + \boldsymbol{b}_s^m),$$

$$p(y^e|\boldsymbol{h}_i^m) = \text{Sigmoid}(\boldsymbol{W}_e^m \boldsymbol{h}_i^m + \boldsymbol{b}_e^m),$$

When $p(y^s|\boldsymbol{h}_i^m) > 0.5$, it indicates that x_i is a candidate start token. Similarly, when $p(y^e|\boldsymbol{h}_i^m) > 0.5$, it indicates that x_i is a candidate end token. After performing the binary classification for all tokens in the text, we obtain the candidate start token set $S_{\text{start}} = \{\ldots, st_i, \ldots\}$ and the candidate end token set $S_{\text{end}} = \{\ldots, ed_i, \ldots\}$, where the superscript m denotes the main task.

After obtaining S_{start} and S_{end}, we take the Cartesian product of the two sets to obtain the candidate entity set with paired start and end tokens $\widetilde{A}^m = \{\ldots, \widetilde{a}_i, \ldots\}$, and apply the following heuristic rules to obtain the final entity set $A^m = \{\ldots, a_i, \ldots\}$: 1) the index of the start token should be smaller than that of the end token; 2) the index distance is not greater than t; 3) the sum of the probabilities of the start and end tokens should exceed the threshold θ; 4) there are at most K candidate entities. We define the loss of entity extraction as:

$$loss_{y_e} = -\frac{1}{B}(\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^s \log(p(y_i^s)) + \hat{y}_i^e \log(p(y_i^e)))),$$

where B is the batch size, n is the number of tokens, $\hat{y}^s$ and $\hat{y}^e$ represent the gold labels for the start and end entity classification tasks, respectively.

Entity Sentiment Classification After obtaining the entity set A^m, we use the entity-context cross-attention module to obtain the context-aware entity representation $\boldsymbol{h}_{a_i}^m$, and then feed $\boldsymbol{h}_{a_i}^m$ into a softmax layer to obtain the sentiment towards a_i:

$$p(y^{a_i}|\boldsymbol{h}_{a_i}^m) = \text{Softmax}(\boldsymbol{W}_a^m \boldsymbol{h}_{a_i}^m + \boldsymbol{b}_{a_i}^m),$$

where $y^{a_i} \in \{\text{positive, negative, mixed}\}$. In this way, we can obtain the set of entities and their corresponding sentiments $AP^m = \{\ldots, (a_i, p_i), \ldots\}$. The loss for entity sentiment classification is:

$$loss_{y_a}^m = -\frac{1}{B}(\frac{1}{|A|}\sum_{i=1}^{|A|}(\hat{y}^{a_i} \log(p(y^{a_i})))).$$

Entity Hierarchical Classification Since the hierarchical entity t_i is composed of multiple entities a, to obtain all hierarchical entities, we first pair each a in the entity set A^m to obtain a set of candidate edges U, and then perform a three-class classification on each candidate edge u_{ij}=(a_i,a_j):

$$p(y^{u_{ij}}|a_i, a_j) = \text{Sigmoid}(\boldsymbol{W}_u[\boldsymbol{h}_{a_i}^m : \boldsymbol{h}_{a_j}^m] + \boldsymbol{b}_u),$$

where $y^{u_{ij}} \in \{0, 1, 2\}$, denoting no relation, hierarchical relation, and coreference relation between entities, respectively. Next, we connect the heads and tails of the candidate edges, remove cycles, and obtain the hierarchical entity set $\{\ldots, t_i, \ldots\}$. The loss for entity hierarchical classification is:

$$loss_{y_u}^m = -\frac{1}{B}(\frac{1}{|U|}\sum_{i=1}^{|U|}(\hat{y}^{u_i} \log(p(y^{u_i})))).$$

4.2.2 Coreference Resolution Modeling

We employ another coreference-RoBERTa model to encode coreference information and incorporate it into the DTSA task. It shares model parameters with the DTSA-RoBERTa model. Considering that both coreference and hierarchical relations are partial order among entities, we use the same classifier to model these two types of relations. Similarly, we obtain a candidate entity set $A^c = \{\ldots, a_i^c, \ldots\}$ and their representations $\{\ldots, h_{a_i}^c, \ldots\}$ for the coreference resolution task. Then, we pair these entities to obtain a set of candidate coreferent entity pairs C, and perform a three-class classification for each candidate entity pair c_{ij} = (a_i^c, a_j^c) to determine whether they refer to the same entity:

$$p(y^{c_{ij}}|a_i^c, a_j^c) = \text{Sigmoid}(\boldsymbol{W}_u[\boldsymbol{h}_{a_i^c}^c : \boldsymbol{h}_{a_j^c}^c] + \boldsymbol{b}_u),$$

where $\boldsymbol{W}_u$ is used to identify both hierarchical and coreference relations. After obtaining the information on whether each candidate entity pair refers to the same entity, we cluster these entity pairs to form the final coreferential clusters. The loss of coreference resolution is:

$$loss_{y_c}^c = -\frac{1}{B}(\frac{1}{|C|}\sum_{i=1}^{|C|}(\hat{y}^{c_i} \log(p(y^{c_i})))).$$

Token-Level Coreference Constraint To incorporate coreference information into the left-path entities while disregarding the noises brought by coreferential entities, we design the token-level coreference constraint (TC) module. By adding token-level constraints on whether the left-path entities and the right-path entities refer to the same entity, we align the representations of the left-path entities with their coreferent entities, thereby enhancing the left-path entities' ability to perceive the contexts of their coreferent entity. Specifically, we utilize the representations of the left-path entities $\boldsymbol{H}_L^m$ and the representations of the right-path

entities $\boldsymbol{H}_L^c$ to obtain a token-level score matrix:

$$p(y_{ij}^{tc}|\boldsymbol{H}_L^m, \boldsymbol{H}_L^c) = \text{Sigmoid}(\boldsymbol{W}_{tc}((\boldsymbol{H}_L^m)^\top \boldsymbol{H}_L^c) + \boldsymbol{b}_{tc}),$$

and the token-level coreference loss is:

$$loss_{y_{tc}}^{tc} = -\frac{1}{B}(\sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{y}^{tc_{ij}} \log(p(y^{tc_{ij}})))).$$

Model Training Our model is based on a multi-task learning framework and consists of three modules: the main DTSA task, and two auxiliary tasks, coreference resolution, and the TC task. The loss is defined as the weighted sum of the individual task losses:

$$loss = loss^m + loss^c + \alpha loss^{tc},$$

where $loss^m$, $loss^c$, and $loss^{tc}$ are the losses for the DTSA, the coreference resolution, and the TC tasks, respectively, and α is the hyperparameter.

5 Experiments

5.1 Datasets and Experimental Settings

Dataset and Metrics The DTSA dataset (Luo et al., 2022) encompasses reviews from four e-commerce domains: Books, Clothing, Hotels, and Restaurants. Each domain contains approximately 1000 annotated documents, with the average number of sentences annotated per document ranging from 3 to 8. Table 2 provides detailed statistics on the dataset division across these domains, distributed in a 7:1:2 ratio for training, validation, and testing, respectively. In evaluation, a target-sentiment pair is viewed as correct if and only if the entities of the target, the sentiment, and their combination are the same as those in the gold target-sentiment pair. We calculate the Precision, Recall, and use $F1$ score (Luo et al., 2022) as the evaluation metric for the DTSA task. In addition, we design a new cluster-based metric allowing for a more accurate assessment of opinion target extraction and report its results in §5.6.

Parameter Setting We initialize the dual-path RoBERTa models with RoBERTa-base parameters. The maximum input length for RoBERTa is set to 512, with the maximum number of entities extracted from the entity set capped at 60. Additionally, the parameters for constraining the entity length t and the entity selection confidence θ are set to 8 and -0.1, respectively. The optimization of model parameters for both paths is conducted using the AdamW optimizer, with a learning rate of 3e-5. The training epochs and dropout rate are set to 30 and 0.1, respectively. During training, model parameters are saved at the point of best performance on the validation set, and the results on the test set are averaged over five random seeds.

Dataset	Train	Dev	Test	Sentences/Doc
Book	690	99	197	5.97
Clothing	649	92	186	3.29
Restaurant	658	94	188	7.91
Hotel	720	103	206	4.19

Table 2: Statistics of the datasets. Sentences/Doc is the average number of sentences per document.

5.2 Baseline Systems

To verify the effectiveness of the coreference information, we adopt the following competitive models as baseline systems, including ChatGPT[1], three generative models, and a non-generative model.

- **GPT-3.5-Turbo**: Experiments are conducted on GPT-3.5-Turbo (Ouyang et al., 2022) with OpenAI's API. Specifically, prompt templates are designed for the DTSA task, and the model's performance is evaluated under a five-shot setting. For specific prompt design, please refer to Table 6.
- **Seq2Seq**: (Luo et al., 2022) uses a generative model to model the DTSA task in an end-to-end manner.
- **BART/T5-Extraction**: The extraction-based generative framework proposed by (Zhang et al., 2021b) for sentence-level ABSA. In this framework, (Song et al., 2023) use BART and T5 as backbones, separating hierarchical entities and sentiments with special symbols, and sequentially outputting the final results.
- **BART/T5-Paraphrase**: The paraphrase-based generative framework proposed by (Zhang et al., 2021a) for sentence-level ABSA. In this framework, (Song et al., 2023) use BART and T5 as backbones, serializing the output sequence of the DTSA task into a natural language sentence.
- **Seq2Struct***: The Encoder-only model proposed by (Song et al., 2023), to reflect the improvement fairly brought by coreference resolution, we replace the backbone of this model with

[1] https://openai.com/chatgpt

Methods	Book	Clothing	Restaurant	Hotel	Average
GPT-3.5-Turbo	16.75	20.10	14.32	20.00	17.79
Seq2Seq (Luo et al., 2022)	34.76	49.40	19.08	34.17	34.35
BART-Extraction	33.83	55.42	33.05	58.90	45.30
BART-Paraphrase	32.90	55.18	33.21	**59.71**	45.25
T5-Extraction	32.66	52.49	32.85	57.92	43.98
T5-Paraphrase	32.64	53.47	33.36	57.95	44.36
Seq2Struct* (Song et al., 2023)	35.55	57.00	38.06	54.24	46.21
Ours	**37.20**	**58.64**	**38.29**	54.46	**47.15**

Table 3: Main results of the DTSA task for our approach and the baseline systems. The best results are in bold.

RoBERTa, remove the GNN module, and replace the entity extraction module with the span-based extraction method used in this paper, while preserving the main structure of Seq2Struct and the modeling approach for each subtask.

5.3 Main Results

The result for the DTSA task is reported in Table 3. There are three noteworthy observations:

Firstly, the average result of the baseline system Seq2Struct* reaches 46.21% after fine-tuning, whereas the result of GPT-3.5-Turbo under a five-shot setting is 17.79%, which is significantly lower than our method. We speculate that the main reasons include two aspects: On one hand, DTSA is a relatively new task with limited related data available, and large models may not have undergone instruction tuning on this type of data. On the other hand, as (Zhang et al., 2023) has pointed out, large models still have significant shortcomings in extracting fine-grained and structured sentiment information.

Secondly, the adapted Seq2Struct* model performs better than other generative baseline models. The average results across the four domains are 0.91 points higher than the best-performing BART-Extraction model. Additionally, in three out of the four domains, this model outperforms the best generative model by 2 points to 5 points, highlighting the effectiveness and adaptability of the Seq2Struct structure. However, the best-performing generative model in the Hotel domain outperforms the results of Seq2Struct* by 5.47 points. One possible reason is that the generative model has a relatively weaker ability to identify entity sentiments compared to Seq2Struct*, and most entity sentiments in the Hotel domain are positive, which reduces the difficulty of entity sentiment identification and leads to better results than Seq2Struct*.

Thirdly, our model with coreference resolution has further improvements based on Seq2Struct*. The average results across the four domains have improved by 0.94 points. Specifically, there are improvements of 1.65 points and 1.64 points in the Book and Clothing domains, respectively. The improvements in the Restaurant and Hotel domains are relatively smaller, at 0.23 points and 0.22 points, respectively. This may be because some entities in the right-path coreference entities are not mentioned in the left path. While introducing coreference information, these entities also affect the accuracy of left-path entity extraction to some extent, resulting in negative effects on the DTSA task. However, these negative effects are mitigated in the Book and Clothing domains by designing the two-path model structure and parameter sharing.

Domains	Seq2Struct*		Ours	
	Coref.	Non-Coref.	Coref.	Non-Coref.
Books	30.19	40.71	**31.16**	**42.76**
Clothing	51.22	**66.00**	**54.03**	65.64
Restaurant	42.30	**64.43**	**43.66**	63.84
Hotel	**36.74**	39.93	35.81	**42.17**
Average	40.11	52.77	**41.16**	**53.60**

Table 4: Results on the dataset with and without Coreference Annotation. The "Coref." indicates labels with coreference annotations, while the "Non-Coref." indicates labels without coreference annotations.

5.4 Evaluation on Test Sets with and without Coreference Annotation

To observe whether coreference information can improve the model's performance on the coreference test set, we divide the test set into two parts: a coreference annotated set and a non-coreference annotated set. We compare and evaluate the results of our model and Seq2Struct* on these two sets. As shown in Table 4, our model achieves an average improvement of 1.05 points and 0.83 points on the coreference test set and non-coreference test set,

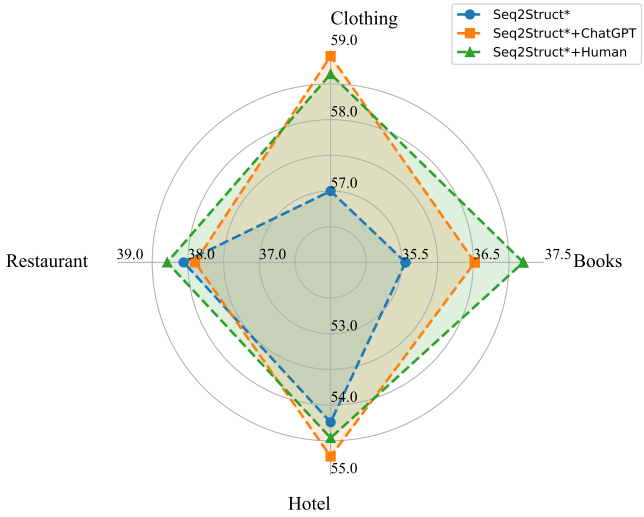

Figure 3: Comparison results between Seq2Struct* with ChatGPT-annotated coreference and human-annotated coreference.

respectively.

Specifically, compared to Seq2Struct*, our model experiences a decrease of 0.93 points on the coreference test set in the Hotel domain. One possible reason is the incorrect extraction of additional coreferential entities, which affects the model's overall performance on the coreference set. Our model also shows improvement on the non-coreference set, possibly because the examples with coreference annotations are more challenging, enhancing the model's ability to recognize coreference while improving its natural language understanding abilities.

5.5 Results with Coreference Annotation Using ChatGPT

To alleviate the high labor and time costs of coreference annotation, we employ ChatGPT to annotate the coreferential entities. The average results of the model with ChatGPT-annotated coreference information in four domains are 47.01%, which is 0.8 points higher than that of Seq2Struct*. Specifically, as shown in Figure 3, Similar to the model with human-annotated coreference information, the model performs better than Seq2Struct* in the Book and Clothing domains, with improvements of 1 point and 1.9 points, respectively. However, in the Restaurant and Hotel domains, the performance is similar, with fluctuations of 0.1 points to 0.3 points. It can be observed that the introduction of entity coreference does not yield significant improvements in the Restaurant and Hotel domains, indicating the need to reduce the impact of other noise (such as additional entities) introduced during the coreference annotation process.

5.6 Evaluaiton with the Coreference-Aware Metric

The opinion targets in the previous ABSA datasets were annotated on separate entities or aspect terms. Due to coreference issues, multiple entities may refer to the same opinion target. For example, in "There are so many different places to choose from in Boston but Paddy Os is by far the best! The bar is spacious with good music...", the "bar" and "Paddy Os" refer to the same opinion target. Therefore, when the model correctly predicts either the "bar" or "Paddy Os", the previous entity-based evaluation metric would consider them as two different opinion targets, leading to inaccuracies in the assessment results. Therefore, based on the original exact matching metric, we consider the coreferential entities and propose the revised coreference cluster-based metric. For the DTSA task, the original Precision and Recall are calculated as follows:

$$P = \frac{\#correct}{\#pred},$$
$$R = \frac{\#correct}{\#gold},$$

where $\#correct_i$ is the number of predicted (t_i, p_i) pairs that are the same as the gold $(\hat{t}_i, \hat{p}_i)$. Since t_i should be an entity of its coreferential cluster, the new metric calculate the similarity between the predicted entity's coreferential cluster and the gold entity's coreferential cluster for $\#correct_i$. Given $\hat{t}_i = \{\hat{a}_1, \ldots, \hat{a}_k\}$, we first require the number of entities in t_i to be equal to k, and calculate the matching score $score(a_j)$ between the coreferential cluster $cluster(a_j)$ of each layer a_j and the coreferential cluster $cluster(\hat{a}_j)$ of $\hat{a}_j$. We then take $score(t_i) = \prod_{j=1}^{k} score(a_j)$, and calculate the current predicted matching score based on $score(t_i)$:

$$\#correct = \sum_{i=1}^{|N|}(\prod_{j=1}^{k} score(a_j)) \times \mathbb{I}(p_i = \hat{p}_i),$$

$$score(a_j) = F1(cluster(a_j), cluster(\hat{a}_j)),$$

where $|N|$ is the number of predicted (t_i, p_i) pairs in the test set with matching scores greater than 0. $F1(cluster(a_j), cluster(\hat{a}_j))$ represents the $F1$ score calculated by treating the coreferential cluster of $\hat{a}_j$ as the gold label.

Domains	Seq2Struct* @Old Metric	Seq2Struct* @New Metric	Ours@Old Metric	Ours@New Metric
Book	35.55	37.15	37.20	38.92
Clothing	57.00	59.24	58.64	60.19
Restaurant	38.06	39.74	38.29	40.44
Hotel	54.24	56.06	54.46	56.23
Average	46.21	48.05	47.15	**48.94**

Table 5: Comparison results of cluster-based metric and entity-based metric on Seq2Struct* and our method.

To mitigate the bias caused by missing coreferential entities during the evaluation process, we assess the impact of the revised metric. As shown in Table 5, the new metric can mitigate the effect of missing coreference targets on Seq2Struct* and our method. Specifically, the new metric results in an improvement ranging from 1.55 points to 2.15 points across different domains, with an average improvement of 1.79 points across all four domains on our method. The improvement brought by the new metric mainly stems from the inclusion of entities in coreference clusters that were not accounted for in the gold labels. These entities are ignored in entity-based metrics but can lead to evaluation errors when they appear in predicted entities. By considering cluster-level matching, this issue can be alleviated.

6 Conclusion

Most research on ABSA focuses on the sentence level. However, DTSA remains an underexplored area. A significant difference between DTSA and sentence-level ABSA is that DTSA requires richer coreference information, necessitating models to possess stronger contextual understanding capabilities. To explore the impact of coreference information on the DTSA task, we annotate coreferential entities with human and ChatGPT and design a multi-task learning framework to verify the positive role of coreference information in DTSA. Additionally, we revise the metrics from exact entity-level matching to a more lenient cluster-level matching to mitigate the bias caused by missing coreferential entities.

Limitations

This paper aims to verify the effectiveness of coreference information on document-level target sentiment analysis, although ChatGPT-annotated coreference information is more efficient and labor-saving compared to manual annotation, it still suffers from the problem of being influenced by prompts. In addition, the revised coreference metrics require manual annotation of coreference information on the test set, which to some extent limits the use of the new evaluation metrics.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by the Natural Science Foundation of China (No. 62076133 and 62006117).

References

Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. Semeval 2022 task 10: structured sentiment analysis. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1280–1295.

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 (ACL-IJCNLP)*, pages 340–350.

Jiahua Chen, Shuai Wang, Sahisnu Mazumder, and Bing Liu. 2020a. A knowledge-driven approach to classifying object and attribute coreferences in opinion mining. *arXiv preprint arXiv:2010.05357*.

Xiao Chen, Changlong Sun, Jingjing Wang, Shoushan Li, Luo Si, Min Zhang, and Guodong Zhou. 2020b. Aspect sentiment classification with document-level sentiment preference modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3667–3677.

Xiaowen Ding and B. Liu. 2010. Resolving object and attribute coreference in opinion mining. In *International Conference on Computational Linguistics*.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240.

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.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. Mvp: Multi-view prompting improves aspect sentiment tuple prediction. *arXiv preprint arXiv:2305.12627*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–515.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 537–546.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th annual Meeting of the association for computational linguistics (ACL)*, pages 151–160.

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.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *proceedings of the 27th international conference on computational linguistics: system demonstrations*, pages 5–9.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 946–956.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 6714–6721.

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. *arXiv preprint arXiv:1907.11692*.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. Doer: Dual cross-shared rnn for aspect term-polarity co-extraction. *arXiv preprint arXiv:1906.01794*.

Yun Luo, Hongjie Cai, Linyi Yang, Yanxia Qin, Rui Xia, and Yue Zhang. 2022. Challenges for open-domain targeted sentiment analysis. *arXiv preprint arXiv:2204.06893*.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv preprint arXiv:1709.00893*.

Dhruv Mullick, Bilal Ghanem, and Alona Fyshe. 2023. Better handling coreference resolution in aspect level sentiment classification by fine-tuning language models. *arXiv preprint arXiv:2307.05646*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27730–27744.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. 2023. Entity-level sentiment analysis (elsa): An exploratory task survey. *arXiv preprint arXiv:2304.14241*.

Nan Song, Hongjie Cai, Rui Xia, Jianfei Yu, Zhen Wu, and Xinyu Dai. 2023. A sequence-to-structure approach to document-level targeted sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7687–7698.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6578–6588.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for end-to-end fine-grained opinion extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2576–2585.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2324–2335.

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2022. An embarrassingly easy but strong baseline for nested named entity recognition. *arXiv preprint arXiv:2208.04534*.

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

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510.

Prompt for coreference cluster annotation
You are an expert in Coreference Resolution of Information Extraction. Given the document, the opinion targets, and their sentiment polarities, you MUST extract ONLY the coreference clusters of the given opinion targets in JSON format. The items in the coreference clusters MUST be the coreferential items of the opinion target. Let's think step by step.
<Document>:
Mark and Roman were amazing hosts. Check-in and check-out procedures were simple. The Studio is very close to Hardvard ave station (green line). I recommend this place.
<Opinion targets and their sentiment polarities>:
{ "place": "Positive", "Check-in": "Positive", "Roman": "Positive", "Mark": "Positive", "Check-out": "Positive" }
<Coreference clusters of the opinion targets>:
{ "place": ["place", "Studio"], "Check-in": ["Check-in"], "Roman": ["Roman"], "Mark": ["Mark"], "check-out": ["check-out"]
other examples
Prompt for target-sentiment pair annotation
You are an expert in Aspect-Based Sentiment Analysis. Given the document, you should ONLY extract the list of target-sentiment pairs. A target-sentiment pair is defined as 'target 1–...–target n##sentiment', where 'target 1–...–target n' is a multi-level opinion target with n denoting the number of its levels, and 'sentiment' in {Positive, Negative, Mixed}. Let's think step by step.
<Document>:
Lovely shoes, and the customer support was wonderful.
<Target-sentiment pairs>:
[shoes##Positive, shoes–customer support##Positive]
other examples

Table 6: Example prompts for coreference cluster and target-sentiment pairs annotation using ChatGPT.

A Entity Coreference Annotation

Coreference Annotation by Humans We utilize the Inception platform (Klie et al., 2018) to annotate coreferential entities. Specifically, for each entity, we first annotate the entities that are coreferential with it and connect these two entities with an undirected edge to represent the coreference relation. In the post-processing stage, we cluster the connected entities into coreferential clusters. These clusters are used for subsequent multi-task training and cluster-based metrics.

Coreference Annotation by ChatGPT To reduce the manual labor and time required for coreference annotation, we employ ChatGPT [2] to annotate the coreferential clusters. First, we construct demonstration examples through five-shot prompting for each domain to extract coreferential entities in JSON format. We tried different task descriptions and output formats to design various prompts. After manually observing and comparing those annotation results from ChatGPT, the final prompt format we selected is shown in Table 6. Each example consists of <Document> and <Opinion targets and their sentiment polarities>. ChatGPT must output <Coreference clusters of the opinion targets> based on the given instruction and the five-shot annotated examples.

To eliminate the noise generated by ChatGPT during the entity extraction process, we remove the articles and adjective possessive pronouns of the entities and merge them as the final version of coreferential clusters. Additionally, we conduct a manual check of 20% of ChatGPT's annotated data (100 documents in each of the five domains), and the assessment yields a coreference annotation accuracy of 92%.

[2] https://openai.com/chatgpt