ChatASU: Evoking LLM's Reflexion to Truly Understand Aspect Sentiment in Dialogues

Yiding Liu, Jingjing Wang*, Jiamin Luo, Tao Zeng, Guodong Zhou
School of Computer Science and Technology, Soochow University, China
No.1, Shizi Street, Suzhou City, Jiangsu Province, China
{20224227068, 20204027003, 20215227014}@stu.suda.edu.cn, {djingwang, gdzhou}@suda.edu.cn

Abstract

Aspect Sentiment Understanding (ASU) in interactive scenarios (e.g., Question-Answering and Dialogue) has attracted ever-more interest in recent years and achieved important progresses. However, existing studies on interactive ASU largely ignore the coreference issue for opinion targets (i.e., aspects), while this phenomenon is ubiquitous in interactive scenarios especially dialogues, limiting the ASU performance. Recently, large language models (LLMs) shows the powerful ability to integrate various NLP tasks with the chat paradigm. In this way, this paper proposes a new **Chat**-based **A**spect **S**entiment **U**nderstanding (ChatASU) task, aiming to explore LLMs' ability in understanding aspect sentiments in dialogue scenarios. Particularly, this ChatASU task introduces a sub-task, i.e., Aspect Chain Reasoning (ACR) task, to address the aspect coreference issue. On this basis, we propose a **T**rusted **S**elf-reflexion **A**pproach (TSA) with ChatGLM as backbone to ChatASU. Specifically, this TSA treats the ACR task as an auxiliary task to boost the performance of the primary ASU task, and further integrates trusted learning into reflexion mechanisms to alleviate the LLMs-intrinsic factual hallucination problem in TSA. Furthermore, a high-quality ChatASU dataset is annotated to evaluate TSA, and extensive experiments show that our proposed TSA can significantly outperform several state-of-the-art baselines, justifying the effectiveness of TSA to ChatASU and the importance of considering the coreference and hallucination issues in ChatASU.

Keywords: Aspect Chain, Hallucination, Aspect Sentiment Understanding, Large Language Models

1. Introduction

Aspect Sentiment Understanding (ASU), a fine-grained sentiment analysis task in the field of sentiment analysis (Liu, 2012; Pontiki et al., 2014), centers on the extraction of aspects from individual sentences and the subsequent prediction of their sentiment polarity (Pontiki et al., 2015; Shen et al., 2018). Throughout the last decade, ASU has garnered widespread utilization across diverse fields, exemplified by its application in e-commerce customer service (Chu et al., 2021) and social opinion mining (Chambers et al., 2015). Recently, some studies focus on the aspect of interactive scenarios, encompassing both single-turn Question-Answering (Wang et al., 2019a) and multi-turn dialogue (Song et al., 2022).

Despite the important progresses achieved by existing studies in ASU, they remain confined to the pre-trained language models (PLMs) phase (Li et al., 2019) and ignore the coreference issue under interactive ASU scenarios. The advent and rapid advancements of large language models (LLMs) like ChatGPT shows the powerful ability to integrate various NLP tasks with the chat paradigm (Zhang et al., 2023). Therefore, to better evaluate the ability of LLMs in understanding aspect sentiments under dialogue scenarios, we propose a new **Chat** **A**spect **S**entiment **U**nderstanding (ChatASU) task and meticulously annotate a high-quality ChatASU dataset (see details in Section 3). In this paper, we believe that our ChatASU task faces two major challenges.

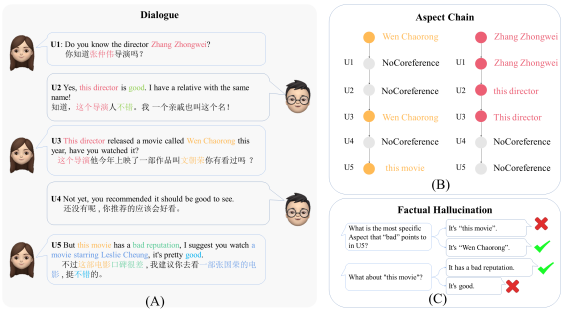


Figure 1: An example to illustrate the coreference and hallucination issue. (A): The concrete dialogue to explain the proposed Aspect Chain and Hallucinations in ChatASU, where different colors represent different aspects. (B): Two Aspect Chains of "Wen Chaorong" and "Zhang Zhongwei" with there corresponding coreference in this dialogue, where NoCoreference means that the current utterance has no coreference. (C): The factual hallucinations exist in ChatASU, i.e., errors in extracting the coreference and predicting the sentiment.

For one thing, how to address the coreference issue for aspects (namely aspect chain issue for short) in dialogues is challenging for LLMs, which could assist in precisely predicting the aspect sentiments. As shown in Figure 1 (B), we can see the aspect chain ("*Wen Chaorong* → ... →*this movie*") within utterances. The aspect "*Wen Chaorong*" is

*Corresponding Author: Jingjing Wang.

LREC-COLING 2024, pages 3075–3085

not referred in utterances **U1**, **U2** and **U4** (i.e., no-conference), but appears in **U3** and **U5** along with "*this movie*" (i.e., coreference). Thus, we denote the aspect "*Wen Chaorong*" in **U3** as *Explicit* aspect and "*this movie*" in **U5** as *Implicit* aspect, where the sentiment *bad* only appears in **U5**, leading to the difficulties of mapping sentiments to aspects. Therefore, a well-behaved approach should consider aspect chain to address the coreference issue and enhance LLMs' ability of understanding aspect sentiments in dialogue scenarios.

For another, LLMs usually exhibit factual hallucination problem in their generative and predictive capabilities (Ji et al., 2023; He et al., 2023). More seriously, due to the existence of aspect chain in ChatASU, LLMs face more serious factual hallucination challenges. Also as shown in Figure 1 (C), a right instance of aspect chain involving "*Wen Chaorong*" is associated with "*this movie*". However, models often misunderstand the context and erroneously link "*this movie*" to "*Zhang Zhongwei*", resulting in the coreference error of factual hallucination issues. Moreover, for the implicit aspect "*this movie*", LLMs tend to predict *good* (in **U2**) instead of *bad* (in **U5**) sentiments. Recently, reflexion provides a way to solve factual hallucination in LLMs. Therefore, a better-behaved approach should consider introducing the trusted learning to further alleviate the factual hallucination challenges of LLMs, thereby enhancing the credibility of LLMs in ASU.

To tackle the aforementioned challenges, we propose a **T**rusted **S**elf-reflexion **A**pproach (TSA) to our ChatASU task. Specifically, we firstly design a chat-style dialogue instruction to input into the LLMs, generating corresponding outputs. Then, we introduce an Aspect Chain Reasoning (ACR) task as an auxiliary task to boost the performance of the primary ASU task, which address the aspect coreference issue. Furthermore, we integrates trusted learning into reflexion mechanisms to alleviate the factual hallucination problem, thereby enhancing the ability of LLMs in understanding aspect sentiments within interactive scenarios. Finally, we employ a reinforcement learning strategy to optimize predictions. Detailed evaluations demonstrate the effectiveness of our proposed TSA. The main contributions of our work are summarized as follows:

- We propose a new ChatASU task with a specially-designed ACR sub-task to address the coreference issue of aspects in dialogue ASU scenarios, which may open up a promising avenue for research in this direction.
- We incorporate both reflexion mechanisms and trusted learning for better understanding aspect chain and alleviating hallucinations problems, thereby enhancing the ability and credibility of LLMs in understanding aspect sentiments.
- We meticulously annotate a high-quality Chinese dataset ChatASU to evaluate the aspect sentiments comprehension ability of LLMs within dialogue ASU scenarios. Our work marks the first of its kind, shedding light on coreference issue in dialogue ASU scenarios and contributing to the evaluation and enhancement of LLMs' performance.

2. Related Work

2.1. Aspect Sentiment Understanding

Aspect Sentiment Understanding (ASU) is a fine-grained sentiment analysis task, which focuses on extracting sentiment information towards specific aspects within the text. Traditional ASU tasks focus on non-interactive scenarios, such as comment text (Peng et al., 2020; Zhang et al., 2021a; Chen et al., 2020b; **?**; Wang et al., 2019c). In recent years, some studies observe the shortcomings posed by non-interactive scenarios and propose ASU tasks based on interactive scenarios, such as Question-Answering scenarios (Wang et al., 2019b), dialogue scenarios (Li et al., 2022; Song et al., 2022), while these studies focus on leveraging pre-trained language models (PLMs). Recently, the emergence of LLMs provides a new paradigm for NLP, which inspires us to explore the capabilities of LLMs in dialogue scenarios. Despite these studies exploring the ASU task, they ignore the issue of coreference, even though this issue is very ubiquitous in dialogue.

Different from the above studies, we propose a new ChatASU task to evaluate the capability of LLM on coreference issue and construct a new dataset to address the coreference issue in dialogues. To our best knowledge, for the ASU task, we are the first study to address coreference issue in dialogue scenarios. In addition, we are devoted to exploring the ability of LLMs to understand dialogues.

2.2. Reflexion Mechanism

Recently, large language models (LLMs) have made a significant impact on various tasks. Although LLMs currently understand the language well, it currently suffers from the hallucination problem (Ji et al., 2023; He et al., 2023). The majority of current studies use reflexion mechanisms to address hallucination problem, such as obtaining the inference path of the LLMs (Wei et al., 2022b), performing actions through observed results (Yao et al., 2023b), searching for problems using a tree structure (Yao et al., 2023a), model editing (Dai et al., 2022), and using heuristic rules to allow the

Split	#Utterances(Dialogues)	#Explicit	#Implicit	Aspect Chain		Quadruple			
				#Max	#Avg	#Pos	#Neu	#Neg	#Total
Train	21612(2400)	8959	6172	11	2.40	7234	472	1261	8967
Valid	2727(300)	1161	770	8	2.45	894	57	180	1131
Test	2723(300)	1146	733	9	2.46	987	71	144	1202

Table 1: The statistics for our annotated ChatASU Dataset. #Explicit denotes the number of explicit aspect entities. #Implicit denotes the number of references towards explicit aspects (e.g., the reference "*this movie*" for the explicit aspect "*WenChaorong*" in Figure 1). #Max and #Avg denote the max length and average length of the aspect chain.

model to reflect (Shinn et al., 2023), etc., where Shinn et al. (2023) inspire our approach.

Different from the above studies, we propose a new Trusted Self-reflexion Approach (TSA) to ChatASU task, which is the first to integrate trusted learning into reflexion mechanisms to alleviate the LLMs-intrinsic factual hallucination problem.

3. Dataset Construction for ChatASU

In order to evaluate the efficiency of the Trusted Self-reflexion Approach (TSA), we construct a new ChatASU dataset based on CASA (Song et al., 2022), which consists of 3000 Chinese dialogues. Since previous studies ignore the coreference issue, this paper defines a new quadruple **"[Explicit Aspect, Implicit Aspect, Opinion, Polarity]"**, which uses explicit aspect and implicit aspect to consider coreference issue in dialogues. Different from existing annotation specifications (Li et al., 2022), this paper does not annotate fine-grained aspect attributes or terms, which can significantly reduce the amount of annotation, and is easy to promote large-scale annotations and applications. It is worth noting that some studies have taken into account coreference between aspects (Chen et al., 2020a), but these studies have not specifically targeted dialogue scenarios. In the following, we will introduce the annotation of explicit aspect, implicit aspect, opinion and polarity, respectively.

Explicit Aspect is used to integrate with **Implicit Aspect** to address the coreference issue. Specifically, we annotate the explicit aspect inside each dialogue based on the following two guidelines.

(1) To simulate a real dialogue environment, if an opinion appears in a sentence, we annotate the most specific aspect entity before current sentence as the explicit aspect. As shown in Figure 1 **U5**, the opinion phrase "bad reputation" appears and the opinion points to the aspect "*Wen Chaorong*" in **U3**. "*Wen Chaorong*" has a coreference expression "*a movie*", but "*Wen Chaorong*" is the most specific aspect, so we annotate "*Wen Chaorong*" as an explicit aspect rather than "*a movie*".

(2) If an aspect has no opinion expression, we do not annotate it. As shown in Figure 1 **U2**, for "*a relative*" there is no expression of sentiment, we do not annotate this aspect.

For Implicit Aspect, we annotate the implicit aspect inside each dialogue based on the following two guidelines.

(1) If an aspect is pronoun of an explicit aspect, we annotate this aspect as an implicit aspect. As shown in Figure 1 **U5**, "*this movie*" is a coreference of the explicit aspect "Wen Chaorong" but not the most specific aspect. Therefore, we annotate "*this movie*" as an implicit aspect.

(2) If a more specific aspect appears after an explicit aspect, we do not modify the previous aspect as an implicit aspect. As shown in **E1** and **E2**, "*Simon Pegg*" is more specific compared to "*her*", while "*her*" comes before "*Simon Pegg*", thus we don't modify "*her*" to be an implicit aspect.
E1 *I like her very much.*
E2 *Her name is Simon Pegg.*

By combining the explicit and implicit aspects, we construct the aspect chain and introduce the ACR task. Specifically, an example of aspect chain ("*Wen Chaorong* $\rightarrow$ ... $\rightarrow$*this movie*") is shown in Figure 1.

Opinion and Polarity is used as the sentiment annotation for the aspect. Specifically, we annotate the opinion and polarity inside each dialogue based on the following three guidelines.

(1) We annotate words or phrases that express explicit sentiment. As shown in Figure 1 **U5**, "*bad reputation*" and "*pretty good*" have explicit sentimental expressions. Therefore, we annotate "*bad reputation*" and "*pretty good*" as opinions and annotate their sentiment polarities as "*negative*" and "*positive*", respectively.

(2) We do not annotate words or phrases with weak sentiment expressions. As shown in Figure 1 **U4**, the phrase "*you recommended it should be good to see*" implies a positive sentiment, while the sentiment expression is not strong enough, thus we do not annotate it.

(3) We categorize opinion as "*positive*", "*negative*", and "*neutral*" based on their sentiment orientation.

During the annotation process, we employ 10 professional annotators. Each dialogue is annotated by two annotators, if they are in disagreement on the annotation result of a dialogue, we employ

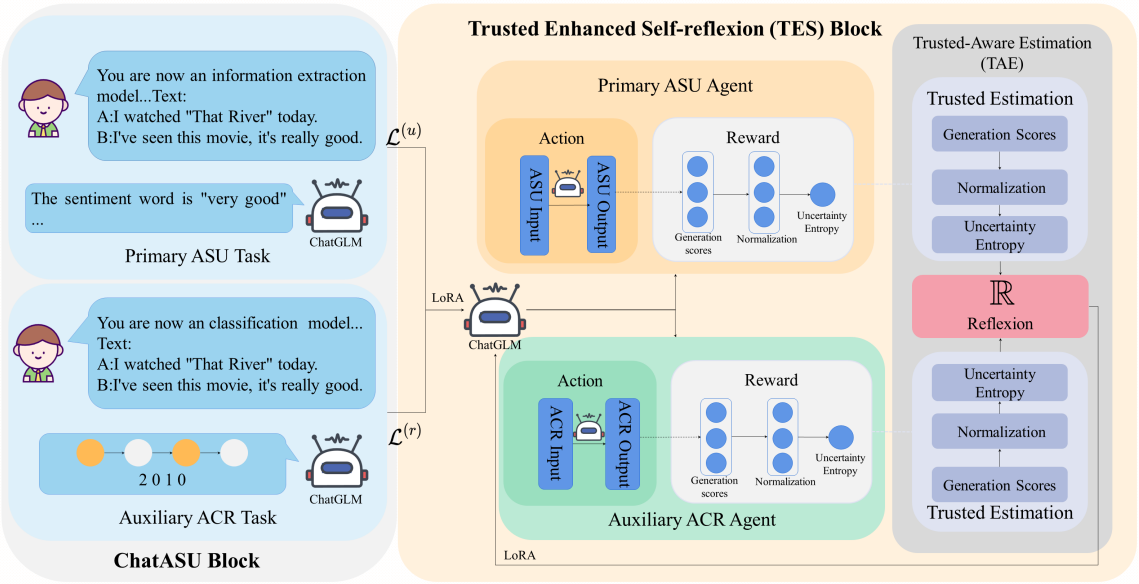


Figure 2: The overall framework of our Trusted Self-reflexion Approach (TSA), consisting of ChatASU Block and Trusted Enhanced Self-reflexion (TES) Block.

an extra domain expert to make the final decision. Finally, we randomly split the dataset into training, validation, and test sets in a ratio of 8:1:1, and the statistics of the dataset are shown in Table 1. Besides, following Cai et al. (2021), we use the matching F1 score and accuracy between two annotators as measures of annotation consistency for the ChatASU dataset. The final F1 score and accuracy are 86.76 and 83.16, respectively.

4. Trusted Self-reflexion Approach

In this section, we formulate the Chat-based Aspect Sentiment Understanding (ChatASU) task, which is consist of two sub-tasks: the primary Aspect Sentiment Understandin (ASU) task and the auxiliary Aspect Chain Reasoning (ACR) task. The specific formulation of the two sub-tasks is described in section 4.1. In this paper, we propose a Trusted Self-reflexion Approach (TSA) for the ChatASU task. The TSA utilizes the ACR task as an auxiliary task to enhance the performance of the ASU task. Additionally, TSA integrates trusted learning into reflexion mechanisms to alleviate the LLMs-intrinsic factual hallucination problem in the ChatASU task. Figure 2 shows the overall architecture of our approach, which is consist of two major parts: **1)** ChatASU Block. **2)** Trusted Enhanced Self-reflexion (TES) Block.

4.1. ChatASU Block

Backbone LLM. A lot of LLMs are open-sourced currently, we consider using ChatGLM-6B (Du et al., 2022) as the backbone. ChatGLM is optimized for Chinese Q&A and dialogues, endowing it with strong Chinese language comprehension abilities. Therefore, we utilize ChatGLM as the backbone in the ChatASU task to explore the capabilities of LLMs in addressing coreference issue.

The ChatASU task is divided into two sub-tasks, the Primary ASU Task and the Auxiliary ACR Task. Their specific formulations are as follows.

Primary ASU Task. Given a dialogue $\mathbf{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_n\}$, where $\boldsymbol{c}_i$ represents the $\boldsymbol{i}$-th utterance and n represents the total number of utterances. The ASU task aims to identify all the quadruples $(\boldsymbol{e}_i, \boldsymbol{r}_i, \boldsymbol{o}_i, \boldsymbol{p}_i)$, where $\boldsymbol{o}_i$ represents opinion in $\boldsymbol{c}_i$ and $\boldsymbol{e}_i$ is the most specific explicit aspect of $\boldsymbol{o}_i$ in $\boldsymbol{c}_j$, $j \in \{1, 2, ..., i\}$. $\boldsymbol{p}_i$ represents the sentiment polarity of $\boldsymbol{o}_i$. Specifically, we need to identify the implicit aspect $\boldsymbol{r}_i$ based on $\boldsymbol{e}_i$ in $\boldsymbol{c}_i$. When no implicit aspect is present, we label it as $\boldsymbol{null}$.

In this paper, we employ the chat-style to fine-tune ChatGLM, enabling ChatGLM to extract quadruples effectively. Inspired by instruct learning (Wei et al., 2022a), we add an "instruct" statement at the beginning of the input sequence and utilize a question-and-answer format to obtain outputs, from which we extract the quadruple. The specific process is described as follows.

• $\mathbf{Input_{ASU}}$. We first present instructions to TSA, which are used to give a clear definition of the task. The *Instruction* is formulated as follows: *"You are now an information extraction model. Please help me to extract opinions from the input and tell me the sentiment polarity of the opinions, what the explicit aspect referred to by the opinion is, and what pronoun is used for the explicit aspect in the utterance where the opinion occurs."*

The input for ChatGLM in the ASU task is obtained as follows. $\mathbf{Input_{ASU}}$ = *Instruction* $[\cdot]$ *text* , where $[\cdot]$ represents string splicing operation, and

text is the dialogue text.

• $\mathbf{Output_{ASU}}$**.** For the output of our approach, we use the way of chat-style like ChatGPT to obtain it. Taking the text "*I watched That River today, the movie is very good to recommend you to see*" as an example, the output format is as follows. $\mathbf{Output_{ASU}}$ = *The opinion is "very good". The sentiment tendency is "POS". The opinion refers to the explicit aspect "That River". The pronoun of "That River" is "the movie".* After getting the output, we filter it to get the target quadruple (*That River*,*the movie*,*very good*,*POS*).

Auxiliary ACR Task. Given an explicit aspect set $\mathbf{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_m\}$, where $\boldsymbol{e}_i$ represents the $\boldsymbol{i}$-th explicit aspect, and m represents the number of explicit aspect in the dialogue. If an utterance contains an explicit aspect or implicit aspect, we label it as 2 or 1, respectively. Otherwise the label is 0. Particularly, if both explicit aspect as well as implicit aspect occur in an utterance, we label it as 2. As shown in Figure 2, the output of ACR task is [2, 0, 1, 0] for a given aspect, which represents that the explicit aspect exists in the first utterance, the implicit aspect exists in the third utterance and no coreference in the second and fourth utterances.

In this paper, we use the chat-style to handle the ACR task and obtain the aspect chain. The input and output formats for the ACR task are as follows.

• $\mathbf{Input_{ACR}}$**.** The input of the ACR task also concatenates instruction and text. The format of *Instruction* is as follows: *"You are now an classification model to judge which utterance in this dialogue appears to be the coreference of* e_i*, outputs 2 if it is an explicit aspect, 1 if it is an implicit aspect, and otherwise 0. Output a sequence of 0, 1, and 2, the length of which is the number of dialogues."*

The input for ChatGLM in the ACR task is obtained as follows. $\mathbf{Input_{ACR}}$ = *Instruction* $[\cdot]$ *text* , where $[\cdot]$ represents string splicing operation, and *text* is the dialogue text.

• $\mathbf{Output_{ACR}}$**.** The output of the ACR task is a sequence consisting of 0, 1 and 2, with the length of the sequence equal to the number of utterances in the dialogue.

4.2. Trusted Enhanced Self-reflexion

Trusted Enhanced Self-reflexion (TES) block integrates trusted learning into reflexion mechanisms by reinforcement learning to alleviate the factual hallucination problem. The TES block comprises two agents: the Primary ASU Agent and the Auxiliary ACR Agent. The specific descriptions of these two agents are as follows.

Primary ASU Agent integrates trusted learning into ASU task to alleviate the factual hallucination problem. The process of ASU Agent in reinforcement learning is as follows. In state s_t, where t represents the t-th time step, we execute action a_t according to policy $\pi(a_t|s_t)$. The specific action and reward of ASU agent are as follows.

• **Action.** We get the ASU task output of ChatGLM. The action is formulated as follows.

$$\mathbf{Output_{ASU}} = \mathrm{ChatGLM}(\mathbf{Input_{ASU}}) \quad (1)$$

• **Reward.** We get the generation scores $\mathbf{Score_{ASU}}$ from the $\mathbf{Output_{ASU}}$, where the generation scores $\mathbf{Score_{ASU}}$ represent the path scores of ChatGLM's beam search in the ASU task. The reward formula for the ASU agent is as follows.

$$\mathbb{R}_{ASU} = \mathbb{R}_{TE}(\mathbf{Score_{ASU}}) \quad (2)$$

where $\mathbb{R}_{TE}(\cdot)$ is described in Eq.(6).

Auxiliary ACR Agent integrates trusted learning into ACR task to alleviate the factual hallucination problem. The process of ACR agent in reinforcement learning is same as ASU Agent. The specific action and reward of ACR agent are as follows.

• **Action.** We get the ACR task output of ChatGLM. The action is formulated as follows.

$$\mathbf{Output_{ACR}} = \mathrm{ChatGLM}(\mathbf{Input_{ACR}}) \quad (3)$$

• **Reward.** We get the generation scores $\mathbf{Score_{ACR}}$ from the $\mathbf{Output_{ACR}}$, where the generation scores $\mathbf{Score_{ACR}}$ represent the path scores of ChatGLM's beam search in the ACR task. The reward formula for the ACR agent is as follows.

$$\mathbb{R}_{ACR} = \mathbb{R}_{TE}(\mathbf{Score_{ACR}}) \quad (4)$$

Trusted-Aware Estimation (TAE) generates rewards through trusted learning and reflexion mechanisms, thereby motivating the ChatGLM to produce credible results. The TAE Block is consist of two parts, Trusted Estimation and Trusted Reflexion.

• **Trusted Estimation (TE)** utilizes the difference in generation scores obtained from beam search as a measure of confidence, encouraging ChatGLM to produce trustworthy results. Specifically, we obtain the output of ChatGLM through beam search and obtain generation scores, denoted as $\mathbf{G} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_n\}$, where n represents the number of generation scores. Next, we enhance the data by normalizing these generation scores.

$$\hat{\boldsymbol{g}}_i = \mathbf{Normalization}(\mathbf{G}) = \frac{\boldsymbol{g}_i - min(\mathbf{G})}{max(\mathbf{G}) - min(\mathbf{G})} \quad (5)$$

where $\boldsymbol{g}_i$ represents the $\boldsymbol{i}$-th generated score, $i = 1, 2, ..., n$. $max(\cdot)$ and $min(\cdot)$ represent the operation of taking the maximum value and the minimum value, respectively.

After obtaining the enhanced results, we use a reward function to calculate the reward. The formal formula for the reward function is as follows.

$$\mathbb{R}_{TE}(\hat{G}) = -\sum_{j=1}^{m} \frac{1}{\sum_{i=1}^{n} \boldsymbol{m}\hat{\boldsymbol{g}}_i \log \hat{\boldsymbol{g}}_i} \quad (6)$$

where $\hat{G} = \{\hat{\boldsymbol{g}}_1, \hat{\boldsymbol{g}}_2, ..., \hat{\boldsymbol{g}}_n\}$. n represents the number of generated scores in an output, and m represents the number of outputs.

As shown in Eq.(6), the reward function is based on the entropy function, but it differs from traditional entropy. In traditional entropy, higher entropy indicates greater disorder, i.e., higher uncertainty and less reliability in the results. However, our reward function operates in the opposite way. When the reward is larger, it means that there is a greater difference in generation scores. We consider that ChatGLM's generation results have smaller uncertainty. In this context, ChatGLM is more confident in the generated results, making the generated outcomes more reliable.

• **Trusted Reflexion** is the final reward function. Inspired by (Shinn et al., 2023), we penalize the ChatGLM when it performs repetition to generate the same result. The final reward integrates trusted learning into reflexion mechanisms to alleviate the LLMs-intrinsic factual hallucination problem. The final reward is formulated as follows.

$$\mathbb{R}=\begin{cases}\alpha\mathbb{R}_{ACR}+\beta\mathbb{R}_{ASU}+\gamma(\mathbb{R}_{Rp}+\mathbb{R}_{Ra}), & \text{if } \boldsymbol{p}=0\\ \alpha\mathbb{R}_{ACR}-\beta\boldsymbol{p}+\gamma(\mathbb{R}_{Rp}+\mathbb{R}_{Ra}), & \text{else}\end{cases} \tag{7}$$

where $\boldsymbol{p}$ is the number of repeat generation, and α, β, γ are hyper-parameters. In reinforcement learning, models often suffer from catastrophic forgetting during training (Fedus et al., 2020). Following Wang et al. (2019a), we introduce the F1 score of ASU task and ACR task (i.e., $\mathbb{R}_{Rp}$ and $\mathbb{R}_{Ra}$) as rewards. The formulas for these two rewards are as follows.

$$\mathbb{R}_{Rp} = F1 = \frac{2 \cdot \mathrm{Pr}_{\mathrm{asu}}\mathrm{Re}_{\mathrm{asu}}}{\mathrm{Pr}_{\mathrm{asu}} + \mathrm{Re}_{\mathrm{asu}}} \tag{8}$$

$$\mathbb{R}_{Ra} = F1 = \frac{2 \cdot \mathrm{Pr}_{\mathrm{acr}}\mathrm{Re}_{\mathrm{acr}}}{\mathrm{Pr}_{\mathrm{acr}} + \mathrm{Re}_{\mathrm{acr}}} \tag{9}$$

where $\mathrm{Pr} = \frac{\mathrm{N_{cp}}}{\mathrm{N_t}}$ and $\mathrm{Re} = \frac{\mathrm{N_{cp}}}{\mathrm{N_p}}$. N_{cp} represents the number of correct predictions. N_t and N_p represent the number of quadruple in label and the number of quadruple in predict, respectively.

4.3. Optimization for TSA

We fine-tune ChatGLM on the ASU task and ACR task using cross-entropy loss. The loss functions for these two tasks are as follows.

$$\mathcal{L}^{(u)} = -\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log(\hat{y}_{ij}) \tag{10}$$

$$\mathcal{L}^{(r)} = -\sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\log(\hat{w}_{ij}) \tag{11}$$

where y and $\hat{y}$ represent the labels and the prediction in the ASU task, respectively. w and $\hat{w}$ represent the labels and the prediction in the ACR task, respectively. N and K represent the length of the label and the length of the vocabulary, respectively. $\mathcal{L}^{(u)}$ and $\mathcal{L}^{(r)}$ represent the loss function of the ASU task and ACR task, respectively. The total loss function is $\mathcal{L}=\mathcal{L}^{(u)} + \mathcal{L}^{(r)}$.

In the reinforcement learning part, we use the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm to maximize our reward. The objective function for the reinforcement learning algorithm is as follows.

$$\max_{\theta} J(\theta)=\max_{\theta}\sum_{\tau} P(\tau;\theta)R(\tau) \tag{12}$$

where $\tau = (s_1, a_1, ..., s_T, a_T)$ is the state-action trajectory, $R(\cdot)$ is the reward function and θ is the parameter of the network.

5. Experiments

5.1. Experimental Settings and Metrics

We empirically evaluate the performance of the TSA on the ASU task using the ChatASU dataset, which is described in Section 3.

We fine-tune ChatGLM-6B[1] using LoRA (Hu et al., 2022)[2]. We set the dimension, scaling factor, dropout rate of the LoRA matrix to be 8, 32, 0.1, respectively, while keeping other parameters at their default values. During fine-tuning, we utilize the Adam optimizer with a learning rate of 2e-4 and weight decay of 5e-4. ChatGLM is trained for 5 epochs on a single A100 40G GPU with a batch size of 2. Training is conducted using Deepspeed[3]. In the reinforcement learning part, the learning rate of ChatGLM is 1e-5 and the batch size is 1. The hyper-parameters α, β and γ are 15, 5 and 3, respectively. The model requires approximately four hours for one training session.

We measure our approach through three perspectives. 1) Single: individual extract of each element. 2) Pair: extract the element pair, i.e., explicit aspect-implicit aspect pair, explicit aspect-opinion pair, implicit aspect-opinion pair. 3) Quadruple Extraction: extract the complete quadruple. Following the prior works (Li et al., 2022), the performance is evaluated with Macro-F1. Moreover, t-test is used to evaluate the significance of the performance difference (Yang and Liu, 1999).

5.2. Baselines

Models like ChatGLM have a much larger number of parameters compared to models like T5. Comparing ChatGLM with models like T5 is unfair. Therefore, we categorize baselines into methods based on Pre-trained Language Models (PLMs)

[1] https://cloud.tsinghua.edu.cn/d/fb9f16d6dc8f482596c2/
[2] https://github.com/liucongg/ChatGLM-Finetuning
[3] https://github.com/microsoft/DeepSpeed

	Approach	Single				Pair			Quadruple
		Explicit	Implicit	Opinion	Polarity	E-O	E-I	I-O	Extraction
PLM	ASQP(Zhang et al., 2021a)	73.08	60.36	61.35	81.52	49.88	50.45	44.59	36.66
	DiaASQ(Li et al., 2022)	49.42	41.7	43.24	56.11	38.61	36.78	34.23	29.09
LLM	ChatGPT(zero-shot)	47.65	55.6	41.88	64.99	27.43	31.05	26.71	22.38
	ChatGPT(In-context learning)	47.82	56.52	43.47	69.57	37.13	34.78	43.48	30.43
	ChatGLM(Du et al., 2022)	67.49	73.70	61.70	82.21	51.32	57.44	52.00	43.49
	Reflexion(Shinn et al., 2023)	68.84	74.11	65.37	86.59	53.32	57.61	53.98	44.62
	TSA	**70.58**	74.45	**65.98**	**86.85**	**55.05**	**58.92**	**54.81**	**46.34**
	- w/o Trusted Learning	69.84	74.22	64.88	86.59	53.66	57.8	53.50	44.88
	- w/o ACR Task	68.98	**75.59**	64.90	86.45	53.14	58.45	54.53	44.98

Table 2: Comparison of several state-of-the-art approaches on ASU task, where "Single" denotes the F1 score extracted separately for each element inside quadruple and "Pair" denotes the F1 score for a pair of two elements inside quadruple. E, I, O denote explicit aspect, implicit aspect, opinion, respectively.

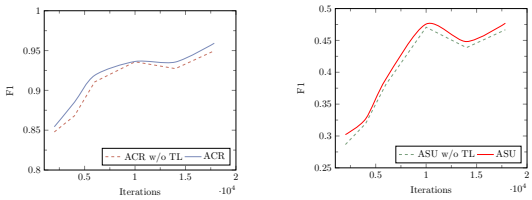
and methods based on Large Language Models (LLMs) according to the number of parameters. **For PLMs**, ASQP (Zhang et al., 2021a) utilizes the T5 (Raffel et al., 2020) to get the output via paraphrasing. DiaASQ (Li et al., 2022) introduces attention mask matrices and combines RoBERTa to model dialogue-specific features via RoBERTa (Liu et al., 2019). Following Zhang et al. 2021b, we use the traditional approach to extract quadruples for these PLMs baselines. **For LLMs**, in ChatGPT, we randomly choose 30 samples and use two ways (zero-shot and in-context learning) to evaluate the capabilities of the ASU task. In zero-shot, we follow the method in Wei et al. (2023) to obtain the results. In in-context learning, we follow the method in Liu et al. (2022) and provide a sample for each input to obtain the results. ChatGLM (Du et al., 2022) directly extracts quadruples through fine-tuning. Reflexion (Shinn et al., 2023) allows the model to self-reflect when its generated results exceed the threshold by setting a threshold value, and this approach is the state-of-the-art approach in the ASU task.

The hyper-parameters of these baselines reported by their public papers are still adopting the same setting, and the others are tuned according to the validation set. We reproduce the approach proposed in Reflexion (Shinn et al., 2023) with ChatGLM as the backbone. To facilitate the corresponding research in this direction, all codes together with datasets will be released via Github[4].

5.3. Experimental Results

Table 2 shows the performance of different approaches to ASU task. From this table, we can see that **1)** The approaches based on LLMs (i.e., Reflexion, TSA) outperform the approaches based on PLMs (i.e., ASQP with T5, DiaASQ with RoBERTa).

[4]https://github.com/Atend9/ChatASU



(a) ACR (b) ASU

Figure 3: (a) The performance of our TSA to ACR task with or without TL during different training steps. (b) The performance of our TSA to ASU task with or without TL during different training steps.

This justifies the powerful comprehension of language by the LLMs. In ChatGPT, due to the model is not open-sourced and only available for interactive use, it results in unsatisfactory performance on ASU tasks. **2)** The models designed for other ASU-related tasks do not perform well in the ASU task. Both ASQP and DiaASQ, based on PLMs and not taking into account the coreference issue, exhibit lower performance than TSA on the ASU task. **3)** The TSA outperforms other LLM-based approaches on the ASU task. Specifically, the TSA outperforms Reflexion in Single, Pair, and Quadruple Extraction by 0.73%, 1.29% and 1.72% on ASU task, respectively. Significance test shows that these improvements are all significant (p-value $<$ 0.05). Particularly, the TSA outperforms GhatGLM in Single, Pair, and Quadruple Extraction by 3.19%, 2.67%, and 2.85% on ASU task, respectively. This justifies the effectiveness of TSA.

To further illustrate the effectiveness of the TSA, we analyze the impact of each part in TSA. Table 2 shows that there is a decrease in our approach performance without the trusted learning and the ACR task. **1)** w/o Trusted Learning (i.e., removing Eq.(5) and Eq.(6)). As shown in Table 2, the performance of TSA in Quadruple Extraction decreases by 1.46%. This indicates that trusted learning is effective in alleviating the problem of factual hal-

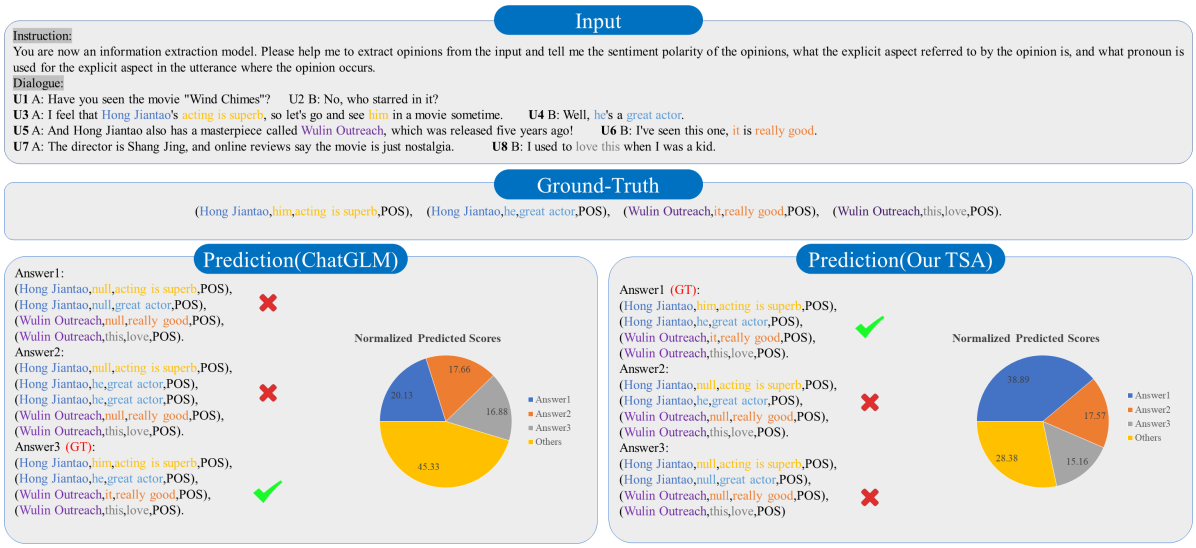


Figure 4: A dialogue example (eight utterances) with their four ground-truth quadruple from the test data of ChatASU dataset. Normalized Predicted Scores denote the normalized predicted percentage for top-3 and "other" generated answers. GT denotes the ground-truth.

lucination. **2)** w/o ACR Task. As shown in Table 2, TSA experiences a decrease in extraction performance by 1.14% in implicit aspect, but it shows an improvement of 1.36% in pair extraction. This indicates that TSA has better relation extraction capabilities in ChatASU and also illustrates the effectiveness of ACR tasks in addressing coreference issues. These further justify the effectiveness of TSA, which again encourages us to consider coreference issue and factual hallucination problem in the ASU task.

6. Analysis and Discussions

6.1. Importance and Robustness Study for ACR Task and Trusted Learning

To verify the importance and robustness of ACR task and trusted learning, we visualize the F1 scores on the ASU task during the training stage of ChatGLM. As shown in Figure 3, we can see that 1) The F1 score of the ACR task grows with the training process, indicating that the ACR task can be efficiently learned by the ChatGLM to address the coreference issue. 2) During the training process, the ACR task and ASU task exhibit similar growth trends. Particularly, between step 15000 and 18000, there is a noticeable improvement in performance on both ACR and ASU tasks. This further validates that the ACR task effectively assists ChatGLM in enhancing its performance on the ASU task. 3) When trusted learning is incorporated into the training process of ChatGLM, there is a significant improvement in performance on both the ACR and ASU tasks. This demonstrates the effectiveness and robustness of trusted learning.

6.2. Qualitative Study

We provide a qualitative analysis of TSA in ASU task on the ChatASU dataset. Figure 4 illustrates samples with coreference issue and factual hallucination problem, showcasing their prediction and normalized predicted scores. We select a dialogue example from ChatASU dataset to analyze the coreference issue and factual hallucination problem. From Figure 4, we can see that 1) ChatGLM fails to recognize the pronoun of "*Hong Jiantao*" in the text, while TSA accurately identifies the pronoun "*him*" referring to "*Hong Jiantao*" in **U4** and "*he*" referring to "*Hong Jiantao*" in **U3**. This demonstrates that TSA can effectively address coreference issue, while ChatGLM exhibits coreference issue. 2) The normalized predicted scores of TSA exhibit significant differences, indicating higher confidence in the generated results. This suggests that the outputs generated by TSA are more reliable. However, in ChatGLM, the differences between the highest prediction scores for Answer1, Answer2, and Answer3 are very small. This implies that ChatGLM lacks confidence in its generated results, making them less reliable. However, TSA enhances the reliability of its generated results through trusted learning.

7. Conclusion

In this paper, in order to address the coreference issue and explore the LLMs' ability in understanding aspect sentiments in dialogues, we introduce a ChatASU task with a specially-designed ACR sub-task and construct a high-quality human annotated dataset for ChatASU. Besides, LLMs currently suffer from the hallucination problem. With these in mind, we propose a Trusted Self-reflexion Approach (TSA), which integrates trusted learning into

reflexion mechanisms to address the coreference issue and alleviate the hallucination problem. Detailed experiments demonstrate the effectiveness of TSA. In our future work, we would like to transfer our approach to mutimodal scenarios (e.g., multimodal aspect-based sentiment analysis) and introduce more information (e.g., eye-contact information) to address the coreference issue in mutimodal dialouges. Moreover, we would like to introduce more reflexion-based approaches (e.g., model editing) to further alleviate the hallucination problem.

Acknowledgements

We thank our anonymous reviewers for their helpful comments. This work was supported by three NSFC grants, i.e., No.62006166, No.62376178 and No.62076175. This work was also supported by a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

8. Bibliographical References

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of ACL/IJCNLP 2021*, pages 340–350.

Nathanael Chambers, Victor Bowen, Ethan Genco, Xisen Tian, Eric Young, Ganesh Harihara, and Eugene Yang. 2015. Identifying political sentiment between nation states with social media. In *Proceedings of the EMNLP 2015*, pages 65–75.

Jiahua Chen, Shuai Wang, Sahisnu Mazumder, and Bing Liu. 2020a. A knowledge-driven approach to classifying object and attribute coreferences in opinion mining. In *Findings of EMNLP 2020*, pages 1616–1626, Online.

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 ACL 2020*, pages 3667–3677.

Zhe Chu, Lu Wang, Yu Run, Ning Ma, Lin Jian, Kun Zhang, and Qiang Liu. 2021. Fine-grained sentiment analysis for meal-to-go scenarios. Https://tech.meituan.com/2021/12/09/meituan-aspect-based-sentiment-analysis-daodian.html.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the ACL 2022*, pages 8493–8502. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of ACL 2022*, pages 320–335.

William Fedus, Dibya Ghosh, John D. Martin, Marc G. Bellemare, Yoshua Bengio, and Hugo Larochelle. 2020. On catastrophic interference in atari 2600 games. *CoRR*, abs/2002.12499.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2023. Exploring human-like translation strategy with large language models. *CoRR*, abs/2305.04118.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Bobo Li, Hao Fei, Fei Li, Yuhan Wu, Jinsong Zhang, Shengqiong Wu, Jingye Li, Yijiang Liu, Lizi Liao, Tat-Seng Chua, and Donghong Ji. 2022. Diaasq : A benchmark of conversational aspect-based sentiment quadruple analysis. *CoRR*, abs/2211.05705.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on EMNLP 2019*, pages 34–41. Association for Computational Linguistics.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114.

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.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *The Thirty-Fourth AAAI 2020*, pages 8600–8607.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of NAACL 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 COLING 2014*, pages 27–35.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Chenlin Shen, Changlong Sun, Jingjing Wang, Yangyang Kang, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. 2018. Sentiment classification towards question-answering with hierarchical matching network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels*, pages 3654–3663.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *CoRR*, abs/2303.11366.

Linfeng Song, Chunlei Xin, Shaopeng Lai, Ante Wang, Jinsong Su, and Kun Xu. 2022. CASA: conversational aspect sentiment analysis for dialogue understanding. *J. Artif. Intell. Res.*, 73:511–533.

Jingjing Wang, Changlong Sun, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. 2019a. Aspect sentiment classification towards question-answering with reinforced bidirectional attention network. In *Proceedings of ACL 2019*, pages 3548–3557.

Jingjing Wang, Changlong Sun, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. 2019b. Aspect sentiment classification towards question-answering with reinforced bidirectional attention network. In *Proceedings of the ACL 2019*, pages 3548–3557.

Jingjing Wang, Changlong Sun, Shoushan Li, Jiancheng Wang, Luo Si, Min Zhang, Xiaozhong Liu, and Guodong Zhou. 2019c. Human-like decision making: Document-level aspect sentiment classification via hierarchical reinforcement learning. 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 5581–5590.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023. Zero-shot information extraction via chatting with chatgpt. *CoRR*, abs/2302.10205.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the ACM SIGIR, 1999, Berkeley, CA, USA*, pages 42–49.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *CoRR*, abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh ICLR 2023*.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of EMNLP 2021*, 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. *CoRR*, abs/2305.15005.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of ACL/IJCNLP 2021*, pages 504–510.