# DEEM: Dynamic Experienced Expert Modeling for Stance Detection

**Xiaolong Wang[*,1,4], Yile Wang[*,2], Sijie Cheng[1,2], Peng Li[✉,2,3], Yang Liu[✉,1,2,3]**
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China
[3]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[4]Jiuquan Satellite Launch Center (JSLC), Gansu, China
wangxl22@mails.tsinghua.edu.cn, wangyile@air.tsinghua.edu.cn
lipeng@air.tsinghua.edu.cn, liuyang2011@tsinghua.edu.cn

**Abstract**

Recent work has made a preliminary attempt to use large language models (LLMs) to solve the stance detection task, showing promising results. However, considering that stance detection usually requires detailed background knowledge, the vanilla reasoning method may neglect the domain knowledge to make a professional and accurate analysis. Thus, there is still room for improvement of LLMs reasoning, especially in leveraging the generation capability of LLMs to simulate specific experts (i.e., multi-agents) to detect the stance. In this paper, different from existing multi-agent works that require detailed descriptions and use fixed experts, we propose a Dynamic Experienced Expert Modeling (DEEM) method which can leverage the generated experienced experts and let LLMs reason in a semi-parametric way, making the experts more generalizable and reliable. Experimental results demonstrate that DEEM consistently achieves the best results on three standard benchmarks, outperforms methods with self-consistency reasoning, and reduces the bias of LLMs.

**Keywords:** Stance Detection, Dynamic Experienced Expert Modeling, Large Language Models

## 1. Introduction

Stance detection (Hasan and Ng, 2014; Küçük and Can, 2020) is a natural language processing (NLP) task that automatically identifies the stance towards a specific target in a given text. For example, the stance of "*Secretary SecPompeo is as corrupt as every other member of the Trump*" is *against* Donald Trump. Such a task has been shown to play an important role in gaining insights into public opinion (Darwish et al., 2017; Lai et al., 2020), understanding political polarization (Darwish et al., 2018), and tracking ideological trends from social media (Conforti et al., 2020).

Recently, large language models (LLMs; Brown et al., 2020; Chowdhery et al., 2022; Ouyang et al., 2022) are developing rapidly and can be applied to various tasks. For example, Zhang et al. (2022) empirically confirm that ChatGPT can achieve impressive performance to detect stance in a zero-shot setting. Zhang et al. (2023a) further improve the results by using chain-of-thought reasoning strategies (Wei et al., 2022). These works have opened up new directions in stance detection.

Despite the success of applying LLMs, conventional reasoning techniques with LLMs could cause hallucinations and factual errors (Guerreiro et al., 2023; Ji et al., 2023), particularly in stance detection. Texts in stance detection usually originate from social media (AlDayel and Magdy, 2021), which are typically short and intricate, necessitating additional domain expertise (He et al., 2022). For example, to detect the stance towards Biden in "*Are you actually trying, as president of the U.S., to start a war??!! #VoteBlueToSaveAmerica2020 #Biden*", we need to know which camp Biden belongs to and the meaning of "#VoteBlueToSaveAmerica2020".

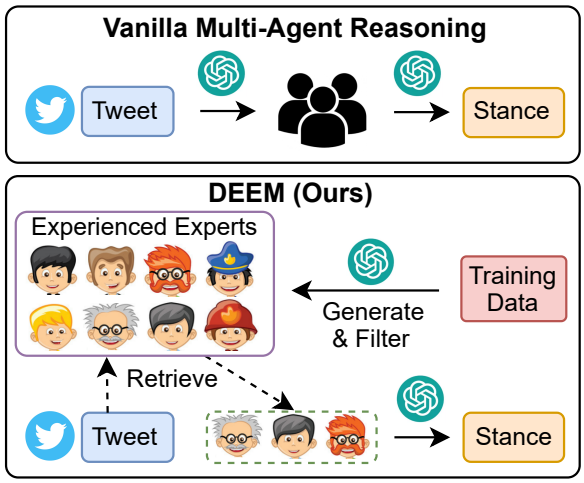



Figure 1: Top: Vanilla multi-agent reasoning for stance detection through generation. Bottom: Our method first generates and filters experienced experts by leveraging training data, then retrieves the related ones during reasoning.

Inspired by *the wisdom of crowds* in sociological theory (Minsky, 1988; Piaget, 2013), we intuitively propose designing multiple capable experts to collaborate in order to come up with a compre-

[*]Equal contribution.
[✉]Corresponding authors.



*LREC-COLING 2024*, pages 4530–4541

hensive stance prediction. Previous studies (Du et al., 2023; Wang et al., 2023d) have attempted to solve reasoning tasks with multi-agent debate and multi-persona self-collaboration. However, their designed agents are generally pre-defined or automatically generated by LLMs, which either require strong prior knowledge or need to be further improved for stance detection tasks. Obviously, predefined agents are fixed, thus it is difficult to adapt to different contexts in social media. Moreover, fully generated agents by LLMs may not be suitable due to the intricate contextualized information, especially in specific domains.

In this work, we propose DEEM, a Dynamic Experienced Expert Modeling method to solve stance detection tasks, as shown in Figure 1. In particular, to better gather the potential expertise for stance detection, we first leverage labeled samples from the existing training data to generate diverse experts. Then, we design two heuristic rules, namely occurrence numbers and response accuracy, to filter the experienced experts and construct an expert repository. Finally, instead of using a fully generative approach, we adopt a dynamic retrieval method to identify relevant experienced experts for new sentences, facilitating discussions for the final prediction.

We evaluate DEEM across both single-target and multi-target stance detection tasks on three widely used datasets, including P-Stance (Li et al., 2021), SemEval-2016 (Mohammad et al., 2016), and MTSD (Sobhani et al., 2017). Experimental results demonstrate that DEEM with dynamic experienced experts can gain substantial improvement across all datasets. Furthermore, it also outperforms reasoning with self-consistency that requires multiple responses and shows potential for reducing the bias of LLMs. Code is available at https://github.com/THUNLP-MT/DEEM.

## 2. Related Work

**Stance Detection**. Early works on stance detection mainly take it as a classification task, leveraging the small language models (Devlin et al., 2019; Nguyen et al., 2020) and learning features from either in-domain or cross-domain training datasets (Augenstein et al., 2016; Zhang et al., 2019; Allaway et al., 2021; Liu et al., 2021; Liang et al., 2022b). With the emergence of LLMs, Zhang et al. (2022, 2023a) first try using ChatGPT to solve the task directly by zero-shot or few-shot reasoning with chain-of-thought, which only requires simple prompts to obtain the political stance from the generated responses. In comparison to their methods, we take inspiration from the multi-agent (Wang et al., 2023a; Xi et al., 2023) and introduce a novel dynamic expert mechanism, enabling LLMs to generate responses from multiple perspectives, providing more comprehensive responses and improve the prediction accuracy.

**Prompt**

What is the attitude of the sentence: "Remind me again how Russian bots ..." to the target "Donald Trump".

Step 1. Select experts based on the sentence:
Cybersecurity_Expert , Social_Media_Expert, Political_Expert.

Step 2. Discussions between experts:
Cybersecurity_Expert: From a cybersecurity perspective, [...]
Social_Media_Expert: As a social media expert, [...]
Political_Expert: From a political science perspective, [...]

The attitude towards Donald Trump is in favor.

(Demonstration)

---

What is the attitude of the sentence: Placeholder to the target "Donald Trump".

Step 1. Select experts based on the sentence:

Figure 2: An example of a few-shot prompt with expert modeling for stance detection. The underlined parts are the sentence, target, and label, respectively. The green texts indicate the manually written experts.

**LLMs Reasoning with Multi-Agent**. Multi-agent strategies have proven to be effective in LLMs reasoning (Talebirad and Nadiri, 2023; Li et al., 2023). By using prompts or instructions that specify the desired role or persona, the model can generate responses based on its understanding of that role and can apply to more complicated scenarios such as social interaction (Park et al., 2023), court simulation (Hamilton, 2023), code development (Qian et al., 2023), and engaging communication games (Xu et al., 2023b). The above works often require detailed role specialization at the beginning of each task (Xu et al., 2023a). In contrast, we propose to discover the experienced personas automatically with the least human effort by using the existing labeled samples, then call for dynamic expertise by retrieving the collected "sentence-expert" pairs. The entire process minimizes the prior knowledge required from stance detection.

## 3. Methods

The overall pipeline of the proposed DEEM is shown in Figure 3. DEEM first generates diverse experts by leveraging the training data. Then it filters the experienced experts that are generalizable and reliable. Finally, it retrieves the experts according to new queried sentences and makes responses.

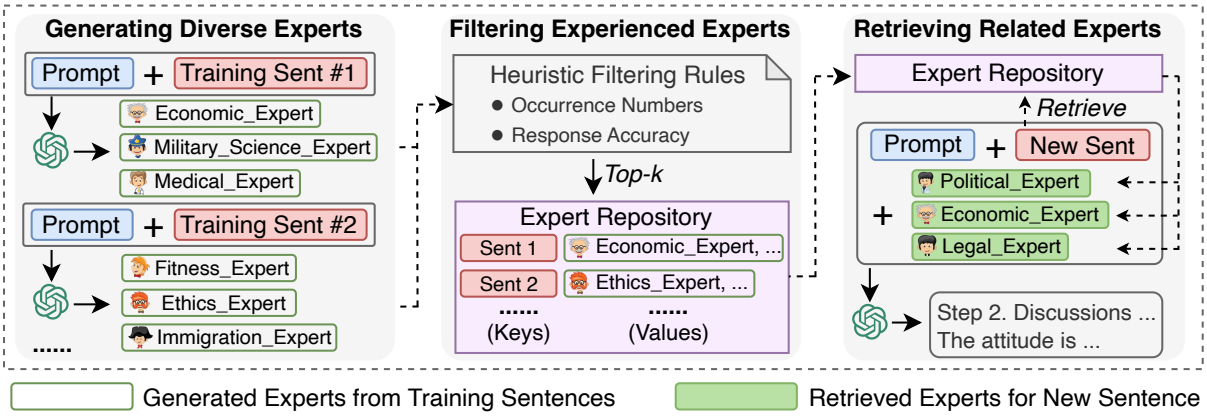



Figure 3: The overall pipeline of our DEEM method. We first use the prompt and training sentences to generate diverse experts (§ 3.1). Then we filter experienced experts according to their occurrence numbers and performance (§ 3.2). Finally, we build the sentence-expert pairs and retrieve the experienced experts for each new sentence (§ 3.3).

### 3.1. Generating Diverse Experts

Conventionally, training datasets are used to fine-tune the parameters of small models. Recently, they can be chosen as prompts for better few-shot in-context learning for LLMs (Liu et al., 2022; Zhang et al., 2023b; Shum et al., 2023). In this paper, we leverage the existing training datasets in a novel way to help generate potentially useful experts for solving stance detection, without using much prior knowledge or detailed role descriptions.

Firstly, given the training dataset with sentence-target-label triplets $\mathcal{D} = \{s_j, t_j, l_j\}_{j=1}^{|\mathcal{D}|}$, we randomly select some of them as held-out ones to construct a prompt as shown in Figure 2:

$$\text{prompt} = s_p \oplus t_p \oplus \mathcal{E}_p \oplus l_p, \tag{1}$$

where $\oplus$ denotes textual concatenation, $s_p$, $t_p$, $l_p$ are the sentence, target, and label of the selected instance in prompt, $\mathcal{E}_p = \{e_p^1, ..., e_p^k\}$ are the *manually written* experts corresponding to the selected instance, such as "Social Media Expert".

Then, we use the LLM $\mathcal{M}$ to generate experts and predicted labels for all other sentences via few-shot in-context learning (Brown et al., 2020):

$$\mathcal{E}_j, \hat{l}_j = \mathcal{M}(\text{prompt} \oplus s_j \oplus t_j), M \tag{2}$$

where $\mathcal{E}_j = \{e_j^1, ..., e_j^k\}$ denotes the *generated* experts, and $\hat{l}_j$ indicates the predicted label to the sentence $s_j$ and target $t_j$.

It is worth noting that more than 1,400 distinct experts can be generated, showing LLMs' strong capability of in-context learning. We further show the detailed expert distributions in Section 5.1.

### 3.2. Filtering Experienced Experts

The generated expert candidates in $\mathcal{E}$ are directly generated by LLMs, which are diverse enough but many do not always match the sentences. To make them more generalizable and reliable to new sentences, we designed two heuristic rules to filter the experienced experts among all candidates.

First, despite a large number of generated experts, many of them only appear a few times. These low-frequency experts are usually from unrelated domains, making it difficult to generalize to new sentences for stance detection. Thus, we require experts who are experienced in stance detection tasks. To fulfill this requirement, the first heuristic rule is the total occurrence numbers $\text{Count}(\cdot)$ of each expert $e_j^m \in \mathcal{E}_j$ in the training dataset as below:

$$\text{Count}(e_j^m) = \sum_{e_i \in \mathcal{E}} \mathbb{1}(e_j^m = e_i), \tag{3}$$

where $e_i$ is the $i$-th element in the collection of generated experts $\mathcal{E}$, and $\mathbb{1}(e_j^m = e_i)$ is the indicator function that equals 1 when $e_j^m$ equals to $e_i$ and 0 otherwise.

Second, it is well known that LLMs can occasionally generate hallucinations (Guerreiro et al., 2023; Ji et al., 2023), thus they could make the responses unreliable and lead to incorrect final predictions. Therefore, experienced experts need to be accurate in analyzing the stance towards the target, thus the second heuristic rule is the total prediction accuracy $\text{Acc}(\cdot)$ of each expert $e_j^m \in \mathcal{E}_j$:

$$\text{Acc}(e_j^m) = \frac{\sum_{e_j^m = e_i} \mathbb{1}(\hat{l}_i = l_i)}{\sum_{e_i \in \mathcal{E}} \mathbb{1}(e_j^m = e_i)}, \tag{4}$$

where $\hat{l}_i$ is the predicted label corresponding to the $i$-th element $e_i$ in Eq. 2, and $l_i$ is the ground-truth label for sentence $s_i$.

Finally, we discard the expert $e_j^m \in \mathcal{E}_j$ who have low prediction accuracy as the threshold (e.g.,

| Method | Including Explanations | Multi-Roles | Verified Experts | Reasoning Type |
|---|---|---|---|---|
| Few-Shot | ✗ | ✗ | - | Gen |
| CoT | ✓ | ✗ | - | Gen |
| Auto-CoT | ✓ | ✗ | - | Re+Gen |
| ExpertPrompt | ✓ | ✗ | ✗ | Gen |
| SPP | ✓ | ✓ | ✗ | Gen |
| DEEM (ours) | ✓ | ✓ | ✓ | Re+Gen |

Table 1: Comparison to typical reasoning methods including Few-Shot, chain-of-thought (CoT), Auto-CoT, ExpertPrompt, and solo performance prompting (SPP). Re: Retrieval. Gen: Generation.

| Datasets | Target | Train | Test |
|---|---|---|---|
| P-Stance | Trump | 6,362 | 796 |
| | Biden | 5,806 | 745 |
| | Sanders | 5,056 | 635 |
| SemEval-2016 | Clinton | 1,898 | 984 |
| | Trump | 2,194 | 707 |
| MTSD | Trump-Clinton | 1,240 | 355 |
| | Trump-Cruz | 922 | 263 |
| | Clinton-Sanders | 957 | 272 |

Table 2: Statistics of P-Stance, SemEval-2016, and MTSD datasets.

$\mathrm{Acc}(e_j^m) < 50\%$). Then we select the rest of the top-$k$ experts (e.g., $k = 10 \sim 30$) according to their occurrence numbers $\mathrm{Count}(e_j^m)$. We take these selected experts $\mathcal{E}'_j$ in each sentence $s_j$ as the final experienced experts. Moreover, we regard all of them as the expert pool $\mathcal{E}' = \mathcal{E}'_1 \cup \cdots \cup \mathcal{E}'_{|\mathcal{D}|}$ to solve the stance detection for a new sentence. We discuss the settings of $\mathrm{Acc}(e_j^m)$ and top-$k$ in Section 5.2.

### 3.3. Retrieving Related Experts

Verified on the training set, all the experts in the expert pool are experienced with both high occurrence and accuracy, these pieces of information can be utilized during the testing phase for retrieval and placed in the prompt as inputs to the model (Guo et al., 2023; Wang et al., 2023c). Overall, to ascertain the stance of a new sentence, it is more effective to choose experienced experts from similar sentences, rather than having LLMs generate potentially unrelated or inexperienced experts directly.

Specifically, we construct a repository to better match the new sentence and these experienced experts. We assign the sentence $s_j$ in the training dataset as the key. Meanwhile, its corresponding filtered experienced experts $\mathcal{E}'_j$ as the value. The resulting sentence-expert pair $\langle s_j, \mathcal{E}'_j \rangle$ indicates that the filtered experienced experts $\mathcal{E}'_j$ can accurately and expertly detect the stance of the sentence $s_j$, thus they have potential to be applied to the similar sentence to $s_j$.

In the inference phase, given a new sentence $s$ and the constructed sentence-expert repository, we can retrieve the top-$h$ related experienced experts according to the textual similarity scores:

$$\mathrm{Sim}(s, s_j) = \frac{\exp(\mathrm{Enc}(s) \cdot \mathrm{Enc}(s_j))}{\sum_{i=1}^{|\mathcal{D}|} \exp(\mathrm{Enc}(s) \cdot \mathrm{Enc}(s_i))}, \quad (5)$$

where $\mathrm{Enc}(\cdot)$ is a sentence encoder, such as SimCSE (Gao et al., 2021).

Finally, we obtain the top-$h$ experienced experts and directly append them to the prompt as shown in Figure 3. Then we use the whole prompt as the input for LLMs to generate the upcoming experts' discussion and the final predicted answer.

### 3.4. Comparison with Other Methods

We compare our method with typical reasoning approaches in Table 1. To involve explanations or specific roles during the reasoning process, CoT (Wei et al., 2022), ExpertPrompt (Xu et al., 2023a) and SPP (Wang et al., 2023d) let LLMs fully generate the explanations or discussions between experts by using the prompt. As for our method, we propose to explore experienced experts from training samples and introduce a retrieval mechanism during the reasoning process.

According to the retrieval mechanism (Borgeaud et al., 2022)„ Auto-CoT (Zhang et al., 2023b) finds similar samples as demonstrations *in the prompt*. In contrast, we retrieve according to constructed "sentence-expert" pairs *during reasoning*, which makes the involved experts more related to the current sentence in a semi-parametric manner.

## 4. Experiments

### 4.1. Experimental Setups

**Datasets.** To evaluate the effectiveness of our method, we comprehensively use three standard stance detection datasets, including both single-target and multi-target tasks: (1) P-stance (Li et al., 2021) is a political stance detection dataset, with each tweet annotated for its stance towards one of three politicians; (2) SemEval-2016 (Mohammad et al., 2016) introduces a shared task on stance detection from tweets, including six targets with one target exclusively for testing; (3) MSTD (Sobhani et al., 2017) is a dataset for multi-target stance detection, primarily focusing on four presidential candidates in the 2016 US election using specific hashtags. Among these datasets, we mainly focus on politicians, such as "Donald Trump". The detailed statistics are shown in Table 2.

**Baselines.** Besides the methods that require supervised fine-tuning, we also compared our method

| Type | Method | P-Stance | | | SemEval-2016 | | MTSD | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | DT | JB | BS | HC | DT | DT-HC | DT-TC | HC-BS | |
| FT | BiCond (Augenstein et al., 2016)♠ | 73.0 | 69.4 | 64.6 | 32.7† | 30.5† | - | - | - | - |
| | BERT (Devlin et al., 2019)♠ | 81.6 | 81.7 | 78.4 | 49.6† | 40.1† | - | - | - | - |
| | BERTweet (Nguyen et al., 2020) | 82.4 | 81.0 | 78.1 | 50.9† | 42.2† | 69.2 | 70.7 | 69.0 | 67.9 |
| | JointCL (Liang et al., 2022b)♡ | - | - | - | 54.8† | 50.5† | - | - | - | - |
| | (`text-davinci-003`) | | | | | | | | | |
| ZS | Zero-Shot (Brown et al., 2020) | 73.8 | 83.3 | 77.5 | 71.8 | 68.3 | 61.6 | 64.7 | 61.4 | 70.3 |
| | DQA (Zhang et al., 2022) | 73.0 | 80.8 | 76.1 | 72.7 | 69.9 | 58.9 | 66.4 | 63.3 | 70.1 |
| FS ($d = 2$) | Few-Shot (Brown et al., 2020) | 79.9 | 85.2 | 78.6 | 79.4 | 73.5 | 68.6 | 65.9 | 70.7 | 75.2 |
| | Manual-CoT (Wei et al., 2022) | 79.3 | 84.9 | 78.4 | 77.2 | 72.5 | 75.0 | 75.6 | 68.8 | 76.5 |
| | StSQA (Zhang et al., 2023a) | 75.2 | 85.2 | 78.9 | 78.3 | 72.3 | 72.6 | 75.9 | 72.0 | 76.3 |
| | Auto-CoT (Zhang et al., 2023b) | 82.9 | 84.7 | 78.4 | 80.7 | 73.8 | 67.9 | 67.4 | 75.3 | 76.4 |
| | ExpertPrompt (Xu et al., 2023a) | 82.8 | 85.5 | 78.7 | 85.2 | 73.0 | 74.1 | 76.8 | 71.5 | 78.5 |
| | SPP (Wang et al., 2023d) | 83.4 | 85.5 | 79.6 | 85.5 | 73.3 | 73.0 | 78.0 | 76.8 | 79.4 |
| | **DEEM** (ours) | **83.7** | **86.0** | **80.4** | **85.7** | **74.8** | **76.5** | **80.1** | **81.3** | **81.1** |
| | Δ (compare w/ second-best results) | +0.3 | +0.5 | +0.8 | +0.2 | +1.0 | +1.5 | +2.1 | +4.5 | +1.7 |
| | (`gpt-3.5-turbo-0301`) | | | | | | | | | |
| ZS | Zero-Shot (Brown et al., 2020) | 83.3 | 82.5 | 79.4 | 79.3 | 71.4 | 73.5 | 67.0 | 73.6 | 76.3 |
| | DQA (Zhang et al., 2022)♠ | 83.2 | 82.0 | 79.4 | 78.0 | 71.3 | 66.2 | 63.2 | 69.3 | 74.1 |
| FS ($d = 2$) | Few-Shot (Brown et al., 2020) | 83.6 | 83.1 | 80.8 | 79.3 | 71.6 | 76.6 | 78.2 | 72.8 | 78.3 |
| | Manual-CoT (Wei et al., 2022) | 85.4 | 83.8 | 80.9 | 79.5 | 71.2 | 77.0 | 77.5 | 76.7 | 79.0 |
| | StSQA (Zhang et al., 2023a)♠ | 85.7 | 82.8 | 80.8 | 78.9 | 71.6 | 77.5 | 78.2 | 81.2 | 79.6 |
| | Auto-CoT (Zhang et al., 2023b) | 84.1 | 82.8 | 80.6 | 84.6 | 73.5 | 77.0 | 76.9 | 76.7 | 79.5 |
| | ExpertPrompt (Xu et al., 2023a) | 84.7 | 84.7 | 81.2 | 83.8 | 77.4 | 80.6 | 77.0 | 79.0 | 81.1 |
| | SPP (Wang et al., 2023d) | 85.1 | 84.6 | 81.5 | 85.3 | 79.5 | 79.5 | 79.8 | 79.8 | 81.9 |
| | **DEEM** (ours) | **86.4** | **86.1** | **82.1** | **85.9** | **80.5** | **81.7** | **80.7** | **83.5** | **83.4** |
| | Δ (compare w/ second-best results) | +0.7 | +1.4 | +0.6 | +0.6 | +1.0 | +1.1 | +0.9 | +2.3 | +2.5 |

Table 3: Main results of baselines and our proposed DEEM. FT: Fine-tuning, ZS: Zero-shot, FS: Few-shot. DT: Donald Trump, JB: Joe Biden, BS: Bernie Sanders, HC: Hillary Clinton, TC: Ted Cruz. †: cross-target setting. ♠: reported by Zhang et al. (2023a), ♡: reported by Liang et al. (2022b). The other results are achieved via our implementation. The best results are in **bold** and the second-best results are underlined.

with recent methods by using LLMs without modifying model parameters:

- DQA (Zhang et al., 2022) uses the template "*What is the attitude of the sentence :* `[Tweet]` *to the target:* `[Target]`. '*favor*' *or* '*against*'." and extract the answers by question answering.
- CoT (Wei et al., 2022) manually provides the explanations in demonstrations and enhances the chain-of-thought reasoning ability of LLMs.
- StSQA (Zhang et al., 2023a) proposes automatic "thought-inducing" and add them to the demonstrations for step-by-step question answering.
- Auto-CoT (Zhang et al., 2023b) automatically selects demonstrations from training data according to semantic diversity.
- ExpertPrompt (Xu et al., 2023a) introduces the identity of experts and customizes information descriptions for LLMs before giving responses.
- SPP (Wang et al., 2023d) proposes solo performance prompting by engaging in multi-turn collaboration with multi-persona during reasoning.

**Metric.** Following Liang et al. (2022a) and Zhang et al. (2023a), we use the $F1_{avg}$, i.e., the average of F1-score on the label '*favor*' and '*against*', as the metric for evaluation and comparison.

**Implementation Details.** We employ InstructGPT (`text-davinci-003`) and ChatGPT (`gpt-3.5-turbo-0301`) as in Zhang et al. (2022, 2023a) through OpenAI API. We set the number of demonstrations $d$ in both the first and third stages as 2. The number of experts (i.e., the number $k$) in the first stage is 3, and the number of top-$h$ in the third stage is also 3. The temperature is set to 0 to ensure the reproducibility of the LLMs' responses.

## 4.2. Main Results

The main comparison results on three standard datasets are reported in Table 3. For fine-tuning models, BERT and BERTweet achieve comparable results on P-Stance, while BERTweet achieves relatively good performance on both SemEval-2016 and MTSD datasets. Moreover, JointCL obtains the highest performance on SemEval-2016. These phenomena show that small models can capture domain knowledge by fine-tuning useful data to further enhance the results.

As for LLMs, zero-shot methods using Instruct-

| Target | Frequency (Proportion) | | | Accuracy (Proportion) | | |
|---|---|---|---|---|---|---|
| | >1% (2.98%) | 0.05-1% (24.32%) | <0.05% (72.70%) | >80% (58.06%) | 50-80% (10.67%) | <50% (31.27%) |
| **Sanders** | Political<br>Ethics<br>Economic | LeaderShip<br>History<br>Technology | Future_Prediction<br>Progmatism<br>Transparency | Immigration<br>Economic<br>History | Media<br>Political<br>Polling | Banking<br>Comedy<br>Alcohol |
| **Trump** | Immigration<br>Political<br>Military_Science | Corruption<br>Social_Policy<br>Taxation | Energy<br>Ethanol<br>Deception | Religious<br>Election<br>Ethics | Political<br>Social_Media<br>Economic | Slang<br>Nationality<br>Endorsement |
| **Clinton** | Gender<br>Healthcare<br>Political | Technology<br>Labor<br>Fashion | Fash_Food<br>Alcohol_Policy<br>Leadership | Ethics<br>Legal<br>National_Security | Unity<br>Gender<br>Political | Geography<br>Diversity<br>Anarchist |
| **Cruz** | Media<br>Political<br>Legal | Unity<br>Technology<br>Polling | Values<br>Nationality<br>Chess | National_Security<br>History<br>Religious_Studies | Political<br>Language<br>Election | Music<br>TeaParty<br>Voting |

Table 4: Distributions and examples of diverse generated experts with different frequencies (Left) and prediction accuracy (Right) to different targets.

GPT can not surpass fine-tuning models in both the P-Stance and MTSD datasets, showing that LLMs with a larger number of parameters do not achieve better results without using specific strategies. Traditional few-shot learning and their reasoning methods obtain significant improvements, especially under InstructGPT, indicating demonstrations and chain-of-thoughts strategies are effective for solving complex tasks. Moreover, the strong performances of ExpertPrompt and SPP prove that using experts is quite useful for stance detection. However, their improvement is not always stable. For example, SPP performs well on P-stance tasks but struggles on SemEval-2016 by using Instruct.

Our method DEEM consistently yields superior results across all three datasets, regardless of whether the InstructGPT or ChatGPT model is utilized. One of the reasons for this superior performance is that DEEM effectively leverages the knowledge within LLMs and adapts it to specific tasks using expert domain knowledge. In some cases, the advantage of DEEM is particularly large (e.g., in the multi-target setting), possibly because it can better capture the underlying structure and relationships within the data. Moreover, compared with expert-based methods, the performance of DEEM is much more stable, indicating that introducing the retrieving module can recall more suitable experts. Overall, DEEM can generalize well and perform better than all other methods, benefiting from dynamic experienced experts.

# 5. Analyses and Discussion

In this section, we conduct a series of analyses to probe the reason behind the effectiveness of our proposed method DEEM. Specifically, we mainly conduct these experiments on the only multi-target stance detection task, i.e., the MTSD dataset.

## 5.1. Expert Distributions

We first investigate expert types according to their frequency and accuracy in the generating stage (Section 3.1). The examples and the distributions of the generated experts are shown in Table 4.

**Frequency.** As for the frequency, we can see that "political experts" appear many times for all targets, showing that LLMs can uncover the shared characteristics of political character and use this to solve the stance detection task. We also find that different targets can exhibit some distinctive types of experts. For example, the "Gender Expert" appears when the target is "Hillary Clinton". Overall, it shows an unbalanced distribution where 72.70% experts have a low frequency, i.e., less than 0.05%. Intuitively, the majority of these low-frequency experts (e.g., "Ethanol Expert", "Chess Expert") do indeed have limited generalizability for stance detection tasks.

**Accuracy.** According to the accuracy, some experts do not show good performance for stance detection. For example, over 30% experts achieve an accuracy of less than 50%. As for the "political experts", the accuracy is mediocre (ranging from 50% to 80%), showing that high frequency does not always lead to high accuracy. Thus we need to combine both the frequency and accuracy to filter experienced ones for better solving stance detection tasks.

## 5.2. Filtering Strategies

We then investigate the accuracy and frequency threshold for filtering experienced experts (Section 3.2), the averaged results for three multi-target pairs are shown in Figure 4.

**Accuracy Threshold.** The impact of the accuracy threshold is shown at the top of Figure 4. The results are relatively bad when the threshold is low (e.g., around 80% when accuracy is lower than

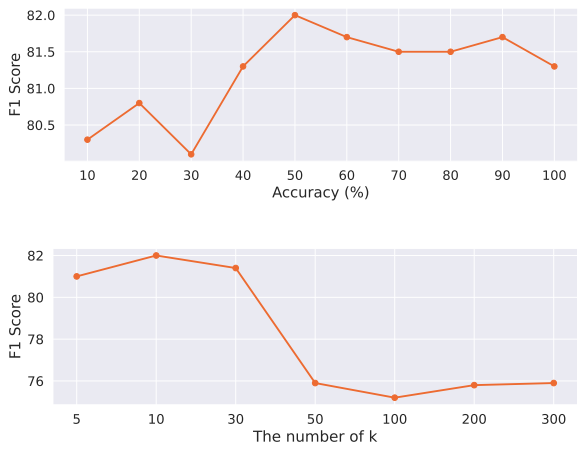



Figure 4: Impact of filtering strategies according to accuracy (Top) and frequency (Bottom).

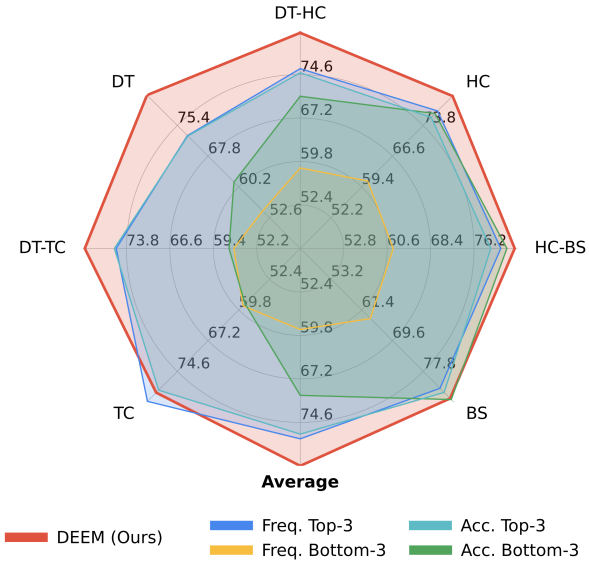



Figure 5: Comparing dynamic experts with fixed experts according to frequency and accuracy.

30%). This shows that the experts with low accuracy may not be generalizable enough. We find that the 50% threshold leads to the best results (around 82%), and the performance does not improve when the threshold continues to increase. The reason can be that an intermediate threshold can maintain both the generalization and diversity of potential useful experts for new sentences, thus showing the best results for test sentences.

**Frequency Threshold.** We set different frequency thresholds, i.e., different top-$k$ selected experts in the expert pool. The results are shown at the bottom of Figure 4. We can see that 10$\sim$30 experts achieve better results. The performance largely reduces when the number of $k$ increases, showing the negative impact of involving the possible unrelated experts with useless information.

## 5.3. Dynamic Experts vs. Fixed Experts

To prove the effectiveness of our dynamic experts mechanism, we compare fixed experts that directly rely on frequency and accuracy for selecting experts. Specifically, we attempt to use the top-3 and bottom-3 experts according to frequency and accuracy. The results are shown in Figure 5.

As we can see, the performance by fixed experts is worse than DEEM with dynamic experts, even if we set the top-3 ones with the highest frequency or accuracy. The performance by using the experts with the least frequency and accuracy further decreases the performance. These findings show that dynamically retrieving the suitably experienced experts according to specific sentences is useful.

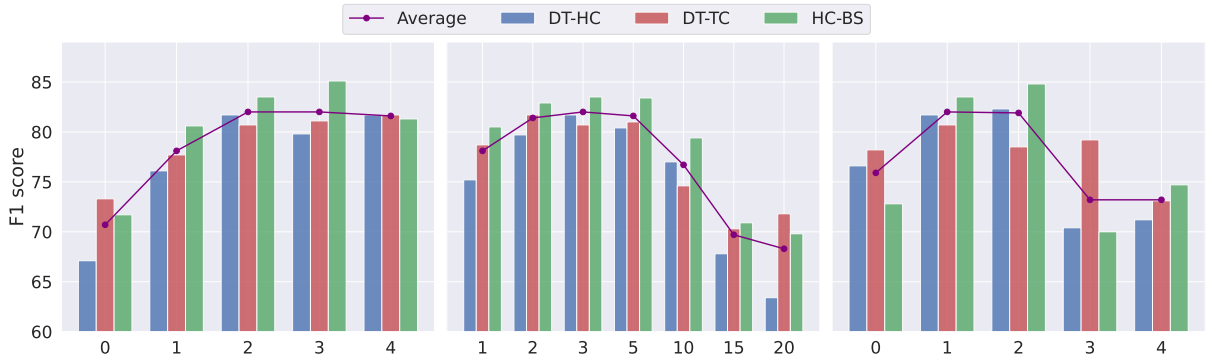
## 5.4. Impact of Demonstrations, Experts, and Discussion Turns

For retrieving the experts and prompting LLMs for predictions (Section 3.3), we investigate the impact of the numbers of demonstrations, retrieved experts, and discussion turns. The overall results are shown in Figure 6.

**Demonstrations.** The results depicted in Figure 6(a) indicate sub-optimal performance during zero- or one-shot reasoning, i.e., when the number of demonstrates is 0 or 1. However, the performance progressively improves and eventually plateaus with the presence of 2 or more demonstrates. This trend suggests that LLMs can deliver commendable performance when provided with exemplars labeled both 'favor' and 'against'.

**Retrieved Experts.** In Figure 6(b), we find that 2 to 5 experts achieve the best performance, and the performance dramatically drops when the experts are 10 or more. This shows that engaging more experts does not consistently lead to better performance, which can be due to introducing noise from unrelated experts, as discussed in Section. 5.2.

**Discussion Turns.** From Figure 6(c), it is apparent that a single turn can already yield satisfactory performance. Additional turns (e.g., 3 or 4) bring improvement. This could be attributed to two primary factors: 1) The difficulty in generating high-quality expert discussions increases with multiple turns, potentially diluting the effectiveness of demonstrations. 2) Complex multi-step reasoning may not be a requisite for stance detection within a sentence.




(a) Number of Demonstrations (b) Number of Experts (c) Number of Discussion Turns

Figure 6: Impact of (a) demonstrations, (b) retrieved experts, and (c) discussion turns during reasoning.

| Method | DT-HC | DT-TC | HC-BS |
|---|---|---|---|
| Few-Shot | 76.6 | 78.2 | 72.8 |
| Few-Shot + SC (N=3) | 76.3 | 80.1 | 76.7 |
| DEEM w/ "Person A/B/C" | 77.2 | 79.0 | 77.0 |
| DEEM w/ "Expert A/B/C" | 78.6 | 78.7 | 76.7 |
| **DEEM** (Ours) | **81.7** | **80.7** | **83.5** |

Table 5: Results of few-shot reasoning, self-consistency (SC) reasoning, and variants of DEEM.

| Method | Trump | Biden | Sanders | Avg. |
|---|---|---|---|---|
| BERTweet | 47.6 | 53.6 | 53.6 | 51.6 |
| DQA | 43.9 | 45.7 | 45.2 | 44.9 |
| StSQA | 54.5 | 55.1 | 53.6 | 54.4 |
| **DEEM** (Ours) | **60.9** | **64.8** | **67.6** | **64.4** |

Table 6: Comparison between methods for the samples in P-Stance dataset with the label "neutral".

### 5.5. Effect of Expert Modeling

To investigate the effect of "experts", we compare self-consistency reasoning and two variants of our proposed DEEM. The results are shown in Table 5.

**Comparison with Self-Consistency Reasoning.** Unlike methods without using experts, our method models three experts and generates the prediction according to their analysis, thus it integrates contextualized information from multiple reasoning paths, which is similar to the self-consistency reasoning method (Wang et al., 2023b). The difference is that self-consistency reasoning requires multiple API calls and multiple responses to get the prediction through voting. From the results we can see that using self-consistency reasoning can improve the few-shot results, showing that the single-reply approach limits the performance. Our method with multi-expert can obtain comparable or better predictions through a single response.

**Comparison with Substitute Roles**. In our method, we specify the profession of the experts, i.e., social media experts, political experts, etc. To make a comparison, we remove the profession and use the substitute roles "expert A/B/C" or "person A/B/C" to involve the multi-role discussion. From the results we can see that our method with specified experts achieves the best results, showing that expert modeling is useful for stance detection, which can offer more reliable results.

### 5.6. Multi-Experts as Bias Reduction

Given the training data distribution, LLMs have shown bias towards specific targets. As Zhang et al. (2023a) demonstrates, ChatGPT tends to show bias towards certain topics with no evident stance, either through direct prompting or chain-of-thought reasoning.

We explore whether the incorporation of multiple experts can mitigate this stance bias. We focus on sentences labeled as 'neutral' and compare predicted stances using different methods. The results are shown in Table 6. For neutral stance samples, the outcomes from DQA and StSQA sometimes fall below or match those from the fine-tuned BERTweet model. This suggests that large models do carry some stance bias, corroborating Zhang et al. (2023a)'s findings. However, our multi-expert method consistently yields the best results for 'neutral' cases, demonstrating the potential of multiple experts in reducing stance bias.

### 5.7. Case Study

One example is given in Figure 7 to show the comparison between the fully generated expert and the retrieved experienced expert. We can see that LLMs misunderstand the text "DOCTOR" and generate the unmatched "Medical Expert", giving incorrect responses with hallucinations. This reflects an important issue with current LLMs in role-playing, which is the inability to generate appropriate characters to solve specific tasks when the roles are not

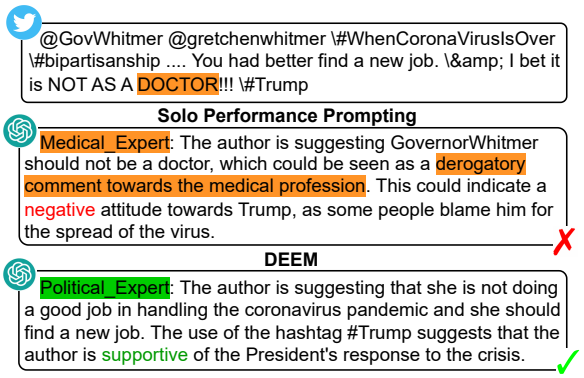



Figure 7: Example of solo performance prompting (Wang et al., 2023d) with generated experts (Middle) and our method with filtered experienced experts through retrieval (Bottom).

provided, especially in non-general domain problems.

In contrast, our DEEM method can determine that this is not a problem related to the medical domain based on the experience of collecting diverse experts, retrieving the experienced "Political Expert" according to this specific question, and showing much more reliable responses in the final.

## 6. Conclusion

We propose DEEM, a dynamic experienced expert modeling method for stance detection. Different from existing multi-agent reasoning methods, DEEM first generates possible diverse experts without leveraging domain knowledge and detailed expert descriptions, then filters the experienced experts to fulfill the generalizability and reliability, and finally it involves a retrieval mechanism during reasoning. Experimental results show that DEEM achieves a consistent improvement over all baselines on three benchmark datasets, outperforming methods with self-consistency reasoning, and reducing the bias of LLMs.

## Acknowledgements

This work is supported by the National Key R&D Program of China (2022ZD0160502) and the National Natural Science Foundation of China (No. 61925601, 62276152, 62306161).

## 7. Bibliographical References

Abeer AlDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4):102597.

Emily Allaway, Malavika Srikanth, and Kathleen McKeown. 2021. Adversarial learning for zero-shot stance detection on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4756–4767, Online. Association for Computational Linguistics.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Will-they-won't-they: A very large dataset for stance detection on Twitter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1724, Online. Association for Computational Linguistics.

Kareem Darwish, Walid Magdy, Afshin Rahimi, Timothy Baldwin, and Norah Abokhodair. 2018. Predicting online islamophobic behavior after #parisattacks. *The Journal of Web Science*, 4(3):34–52.

Kareem Darwish, Walid Magdy, and Tahar Zanouda. 2017. Trump vs. hillary: What went viral during the 2016 us presidential election. In *Social Informatics*, pages 143–161, Cham. Springer International Publishing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Zhicheng Guo, Sijie Cheng, Yile Wang, Peng Li, and Yang Liu. 2023. Prompt-guided retrieval augmentation for non-knowledge-intensive tasks. *arXiv preprint arXiv:2305.17653*.

Sil Hamilton. 2023. Blind judgement: Agent-based supreme court modelling with gpt. *arXiv preprint arXiv:2301.05327*.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762, Doha, Qatar. Association for Computational Linguistics.

Zihao He, Negar Mokhberian, and Kristina Lerman. 2022. Infusing knowledge from Wikipedia to enhance stance detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 71–77, Dublin, Ireland. Association for Computational Linguistics.

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

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1):1–37.

Mirko Lai, Alessandra Teresa Cignarella, Delia Irazú Hernández Farías, Cristina Bosco, Viviana Patti, and Paolo Rosso. 2020. Multilingual stance detection in social media political debates. *Computer Speech & Language*, 63:101075.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large scale language model society.

Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. P-stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2355–2365, Online. Association for Computational Linguistics.

Bin Liang, Zixiao Chen, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2022a. Zero-shot stance detection via contrastive learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2738–2747, New York, NY, USA. Association for Computing Machinery.

Bin Liang, Qinglin Zhu, Xiang Li, Min Yang, Lin Gui, Yulan He, and Ruifeng Xu. 2022b. JointCL: A joint contrastive learning framework for zero-shot stance detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

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*

*(DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Rui Liu, Zheng Lin, Yutong Tan, and Weiping Wang. 2021. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3152–3157, Online. Association for Computational Linguistics.

Marvin Minsky. 1988. *Society of mind*. Simon and Schuster.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

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*, 35:27730–27744.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.

Jean Piaget. 2013. *The construction of reality in the child*, volume 82. Routledge.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

KaShun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. *arXiv preprint arXiv:2302.12822*.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557, Valencia, Spain. Association for Computational Linguistics.

Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023a. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *ICLR 2023*.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023c. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore. Association for Computational Linguistics.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023d. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023a. ExpertPrompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023b.

Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*.

Bowen Zhang, Daijun Ding, and Liwen Jing. 2022. How would stance detection techniques evolve after the launch of chatgpt? *arXiv preprint arXiv:2212.14548*.

Bowen Zhang, Xianghua Fu, Daijun Ding, Hu Huang, Yangyang Li, and Liwen Jing. 2023a. Investigating chain-of-thought with chatgpt for stance detection on social media. *arXiv preprint arXiv:2304.03087*.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4568–4578, Hong Kong, China. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*.