Pattern Recognition Letters 197 (2025) 81–87

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Improving LLM-based opinion expression identification with dependency syntax

Qiujing Xu [a], Peiming Guo [b], Fei Li [a], Meishan Zhang [b], Donghong Ji [a,*]

[a] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China

[b] Institute of Computing and Intelligence, Harbin Institute of Technology (Shenzhen), China

ARTICLE INFO

Editor: S. Sarkar

Keywords:
Opinion expression identification
Large language models
Dependency syntax

ABSTRACT

Opinion expression identification (OEI), a crucial task in fine-grained opinion mining, has received long-term attention for several decades. Recently, large language models (LLMs) have demonstrated substantial potential on the task. However, structural-aware syntax features, which have proven highly effective for encoder-based OEI models, remain challenging to be explored under the LLM paradigm. In this work, we introduce a novel approach that successfully enhances LLM-based OEI with the aid of dependency syntax. We start with a well-formed prompt learning framework for OEI, and then enrich the prompting text with syntax information from an off-the-shelf dependency parser. To mitigate the negative impact of irrelevant dependency structures, we employ a BERT-based CRF model as a retriever to select only salient dependencies. Experiments on three benchmark datasets covering English, Chinese and Portuguese indicate that our method is highly effective, resulting in significant improvements on all datasets. We also provide detailed analysis to understand our method in-depth.

1. Introduction

Opinion expression identification (OEI), which aims to recognize opinion expressions as well as their polarities, is one core task in fine-grained opinion mining [1–6]. The task can provide key information regarding sentiments and opinions to facilitate downstream processing in natural language processing (NLP) [7–10]. Fig. 1 shows an illustrated example. OEI has been researched intensively for decades, ranging from the early statistical machine learning models [11], neural-based encoder–decoder models [4,12,13], as well as generation-style decode-only methods empowered by large language models (LLMs) [14]. With the aid of well pretrained language models, the task performance has reached at a high level due to the strong capacity in semantic representation and inference.

Among these models, the LLM-based generation paradigm is one of the most promising, providing state-of-the-art results across a wide range of NLP tasks [15–17]. Our OEI task is no exception [14]. By employing straightforward prompt learning on well-pretrained LLMs, we are able to attain highly competitive performance on OEI. Despite the success, the LLM-based generation architecture struggles to explore information regarding structural forms, which makes the paradigm insufficient when these types of information are important. OEI is precisely the unfortunate case, because structural-aware syntax information has been demonstrated highly valuable under the previous encoder–decoder paradigm [6,13,18].

In this work, we present the first attempt to integrate structural-formed syntax information into LLM-based OEI. Here, we take dependency syntax as the pilot study. Our key point lies in two problems. First, how to linearize dependency parsing trees into narrative texts, so that the dependency information can be integrated directly with LLM prompting. Second, how to keep only the salient dependency information for LLM prompting, since the LLM reasoning is prone to redundant texts while the above linearization can produce very long contexts. Both problems are essential in order to make the dependency syntax positive for our LLM-based OEI.

With regard to the first problem, a full dependency tree is converted into a collection of descriptive texts by applying a predefined template over the inside dependencies. A further step is to order these dependency descriptions for full linearization. Here we address the issue along with the second problem. For the second problem, we use a standard BERT-based OEI model to weight all dependencies in the full parsing tree, and thus each dependency description is naturally prioritized. Alternatively, the BERT-based model can also be regarded as a

* Corresponding author.
E-mail addresses: xuqiujing@whu.edu.cn (Q. Xu), 24b351013@stu.hit.edu.cn (P. Guo), lifei_csnlp@whu.edu.cn (F. Li), zhangmeishan@hit.edu.cn (M. Zhang), dhji@whu.edu.cn (D. Ji).

https://doi.org/10.1016/j.patrec.2025.07.012
Received 22 December 2024; Received in revised form 30 April 2025; Accepted 12 July 2025
Available online 24 July 2025

Fig. 1. An illustrated example for OEI. The red rectangle with solid lines indicates a positive opinion, and the green dashed one indicates negative.

dependency retriever that ranks and filters dependencies jointly, and the overall framework actually follows the line of retrieval augmented generation (RAG) of exploring LLMs [19,20].

We conduct experiments on two benchmark datasets to verify our method, covering both English and Chinese. First, we check the LLM's capability in dealing with opinion expression identification. The results are consistent with our expectation that LLMs can deliver very competitive performance. Second, we evaluate the model performance after dependency syntax integration. Indeed, if feeding the linearized dependency syntax directly without filtering, negative results would be shown. Finally, through our proposed weighting strategy, we can obtain significant improvements over the corresponding LLM baseline. We also perform several in-depth analyses to better understand the mechanism of our method. All our source codes as well as the datasets are publicly available at https://github.com/xuqiujingqzy/LLM-OEI-Syntax to facilitate reproduction for research purposes.

2. Related work

Opinion mining has been a hot topic for decades [2–4,6,21–25]. OEI is a critical task in fine-grained opinion mining [1]. Initially, the task is modeled as a sequence labeling problem with human-crafted discrete features for encoding and CRF for decoding [11,26–28]. The later investigations show that neural networks are much better for feature representation [12,13,29]. Furthermore, this advantage has significantly grown with the advent of pretrained language models. The BERT-based CRF architecture has been the prevalent approach to this task [30], and latterly there are several other sophisticated structural learning encoder–decoder architectures backboned with BERT [23–25, 31]. Recently, the success of LLMs has led to a paradigm shift across a wide range of NLP tasks [15–17,32]. OEI has also been greatly promoted, outperforming the previous BERT-based CRF models [14]. To date, Jia et al. [14] is the only work of OEI using LLMs.

Syntax information has been highly useful for various NLP tasks [33–36], including relation extraction [33], pretrained language model [35,36], semantic representation [34] and opinion expression identification [6,13]. In the era of statistical machine learning models, syntax is manually converted to symbolized discrete features [33,37]. The syntax information is also intensively examined in neural network models for these tasks [34,38,39]. Concretely, Wu et al. [40] inject dependency syntax into a neural transition system for opinion role labeling. Graph neural network encodes structure and semantic information from dependency trees to enhance word representations and improve opinion extraction performance. Xia et al. [39] explore two different methods, multi-task learning and graph neural network, to integrate constituency syntax into a neural span-based model for opinion mining task. The former trains a joint span classification model for opinion recognition and constituency parsing. The latter employs graph convolutional network to extract syntax knowledge for better text representations. The development of syntax-aware word representations has been investigated in a number of studies, which are then incorporated into neural NLP models as general backbones [35,36,41, 42]. Due to its concise formalization, dependency syntax has received relatively-more attention than others [43–45].

While generative LLM-based models have become increasingly popular nowadays [15,17,32], syntax knowledge integration has been neglected unexpectedly in this new paradigm. The reason of this situation is that previously well-worked methods of external knowledge integration on encoder–decoder models [46] are unadaptable to the prompt-based decoder-only paradigm. We are unable to build an extra effective encoder, such as graph neural network, modeling dependency syntax under the LLM framework. To address the problem, we may direct train syntax-aware (or knowledge-aware) LLMs [47], which might be practical for middle-size language models below 1B sizes but apparently very expensive for LLMs. Alteratively, we can equip the prompt text with syntax information, which is the idea of this work. However, reducing the negative impacts of redundant information is a very difficult task for LLM prompting [48]. In fact, our method can be regarded as an RAG exploration of LLMs [19,20]: (1) retrieve salient dependency syntax knowledge, (2) augment the knowledge to help the generation of LLM-based OEI. To our knowledge, this is also the first attempt of using dependency syntax under the generative LLM-based paradigm.

3. Method

Dependency syntax is one source of highly-effective information previously [6,13,39], while the same information has been very seldom exploited in the LLM-based paradigm. The main reason lies in that the exploration of structured features is very difficult in the generation-style prompt learning framework. Here we use a two-stage strategy to let our LLM-based OEI model exploit dependency syntax effectively, as shown in Fig. 2. In the following, we will first introduce the task definition of opinion expression identification, then describe our baseline LLM-based OEI model, and finally propose our method of integrating dependency syntax for our baseline.

3.1. Task definition

Given an input text $x = w_1 w_2 \cdots w_n$ (n is the length of the sentence), opinion expression identification is to recognize all attitude expressions and their polarities, i.e. $y = \{\langle \text{exp}_1, p_1 \rangle, \ldots, \langle \text{exp}_1, p_1 \rangle\}$, where $\text{exp}_i = w_{s_i} \cdots w_{e_i} (s_i \leq e_i)$ indicates one opinion expression and $p_i \in \{\text{POS}, \text{NEG}\}$ indicates its polarity (positive or negative). Here, we assume that there is no overlapped opinion expressions following previous studies. The task can be treated as a sequence labeling problem under the encoder–decoder setting [11,27,28], where the output labels are {B-POS, I-POS, B-NEG, I-NEG, O} (B indicates begin of an expression, I indicates inside, O indicates other). Under the LLM-based architecture, we always use the strong generation capability of LLM, outputting a textual response in which all opinion expressions in text are enclosed with specific predefined symbols [14]. For each opinion $\text{exp}_i = w_{s_i} \cdots w_{e_i}, p_i$, we insert two symbols: $\langle p_i \rangle$ and $\langle / p_i \rangle$, in front and at the end of the expression, respectively (i.e., $\langle p_i \rangle\ w_{s_i} \cdots w_{e_i}\ \langle / p_i \rangle$).

3.2. Baseline: LLM-based OEI

Auto-regressive large language models have received considerable attention in the NLP community recently. A range of NLP tasks obtained significant improvements with the support of LLM by prompt learning, due to the strong reasoning capability emerging from large language models. We illustrate our baseline model in Fig. 2(b) without the blue text, which is almost the model of Jia et al. [14] without the speech part. As mentioned above, we follow the standard method by inserting OEI symbols into the original text to convert it into a generation problem, and use an improved prompt for better handling OEI of a certain language.

Our prompt text consists of three parts: {Characterization} {Task Description} {Input}. {Characterization} (i.e., You are a native English speaker and are capable of recognizing subtle expressions of opinion in text) activates the role of the large language model in the opinion expression identification task. Following, we introduce task goal in {Task Description} by the sentence: Your goal is to annotate all opinion expressions inside a text accurately. Finally, the text including opinion expressions to be extracted is appended into the prompt as {Input}

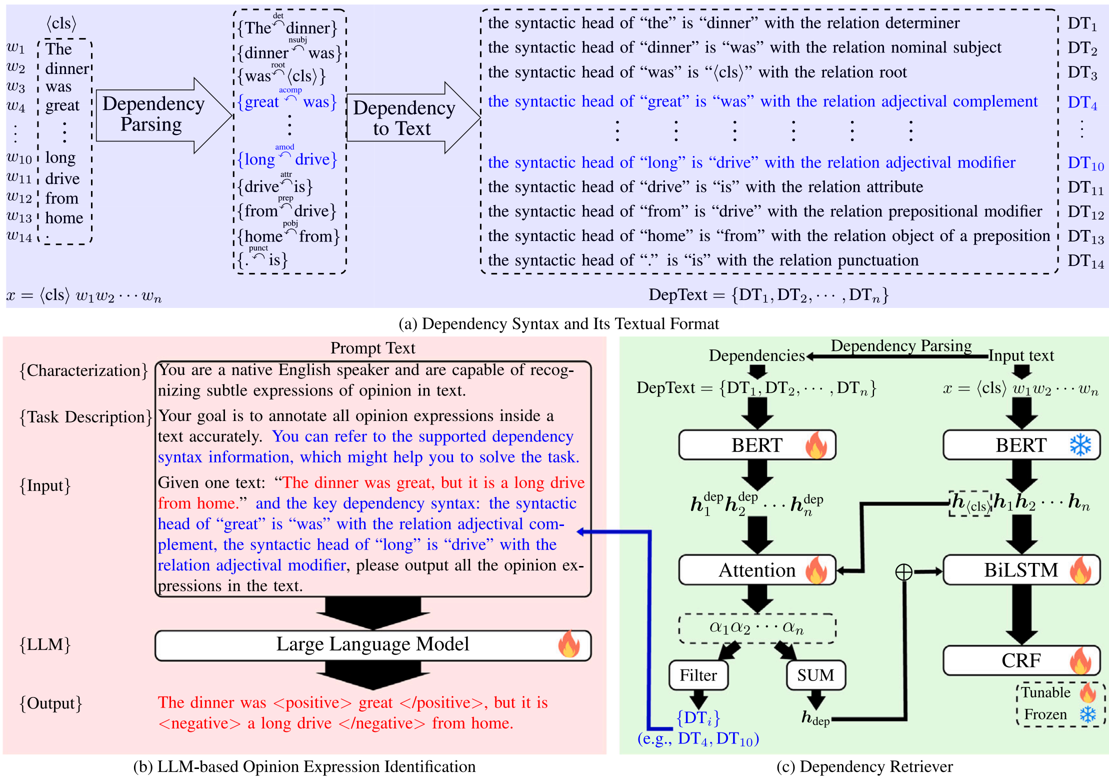

Fig. 2. An example to illustrate the overall architecture of our method. We exploit a two-stage architecture to integrate dependency syntax into the LLM-based OEI model. We first employ a representative BERT-based OEI model as a dependency retriever to retrieve important dependencies. And then feed the selected dependencies by their importance order to enrich the prompt text of the baseline LLM-based OEI model. ⟨cls⟩ is the text start marker in BERT which is usually used for full-text representation.

(e.g., Given one text: "The dinner was great, but it is a long drive from home"., please output all the opinion expressions in the text.). With the prompt text being fed into a well-pretrained large language model, a standard response, which encloses the OEI results {Output}, is generated accordingly. As shown in Fig. 2, we enclose opinion expressions by special identifiers, including <positive>, </positive>, <negative> and </negative>. In the post-processing process, these special tokens can identify the boundaries and sentiment polarities of opinion expressions.

We can optimize the pretrained LLM for our OEI task further, aligning with the supervised learning setting. By converting the OEI training datasets into input–output pairs of LLM prompt learning (i.e., <prompt instruction, output text>), the backbone LLM can be fine-tuned as a standard instruction tuning process. The mechanism has been widely adopted in adaption of LLMs to a specific NLP task or domain, which can lead to much better performance in most cases. We exploit LORA-tuning to better fit large language models to our OEI work.

3.3. Dependency syntax integration

Syntax information has been extensively exploited for opinion mining, including OEI [39,40]. Dependency syntax is the most popular feature source due to its concise structure and rich data resources. One can easily feed the dependency-structure information into any type of encoder–decoder models, while the key of dependency syntax injection into decoder-only LLMs is the prompt text. The most straightforward thing here is to transform the dependency syntax tree into a flat text and then combine it with the original prompt. The transformation, however, is inevitably associated with information losses. Fig. 2(a) shows the transformation method.

As a preparation, first we decompose a full dependency tree from dependency parsing into individual dependencies, which are triples including head words, dependent words and dependency labels. Then, we transform each dependency into a sentence, resulting in a set of dependency descriptions DepText $= \{\mathrm{DT}_1, \mathrm{DT}_2, \ldots, \mathrm{DT}_n\}$(Each word w_i would produce one dependency exactly). The tree-to-dependency process can be mutually convertible, whereas the dependency-to-text transformation can be achieved by a template with very little information loss: The syntactic head of w_i is w_{h_i} with the relation l. (i.e., a triple $\langle w_i, w_{h_i}, l\rangle$).

There are still two issues to be addressed. Firstly, it is necessary to determine the order of the aforementioned dependency descriptions in order to concatenate them, where the most reasonable strategy would be to rank them in order of their importance to opinion expression identification. Secondly, the simple joining of all dependency descriptions may result in too long a prompt text for LLM reasoning. Assuming the input text length being n, the length of the prompt syntax text could be at least $\|\text{template}\| * n$ (here $\|\text{template}\| = 12$). In practice, keeping only the salient dependencies is preferable. We examine the influence of the number of retained dependencies on the final opinion expression identification performance in the following experiment section.

Here we exploit a two-stage architecture, i.e., (1) dependency retrieve and (2) LLM-based reasoning, to solve the above two issues concurrently. In fact, this architecture naturally aligns with the RAG mechanism which has been widely adopted to enhance LLM-based models [19,20]. In this work, we exploit a representative BERT-based OEI model as a dependency retriever to solve both issues simultaneously, and then feed the selected dependencies by their importance order to enrich the prompt text of our baseline model.

Concretely, we adopt a BERT-based CRF model to help the dependency selection. The retriever details are shown in Fig. 2(c). As depicted, we extend a standard BERT-BiLSTM-CRF model with a dependency representation module to accomplish the goal. We condense the dependency syntax tree of the input text (i.e., $x = \langle\text{cls}\rangle\ w_1\ w_2\ \cdots\ w_n$, where ⟨cls⟩ is a special start marker in BERT) into a vector representation (i.e., $\boldsymbol{h}_{\text{dep}}$), and then concatenate it with the original BERT representation $\boldsymbol{h}_{\langle\text{cls}\rangle}\ \boldsymbol{h}_1\ \boldsymbol{h}_2\ \cdots\ \boldsymbol{h}_n$ of $x = \langle\text{cls}\rangle\ w_1\ w_2\ \cdots\ w_n$, which are finally fed into the BiLSTM network, forming a simple dependency-aware opinion expression identification model.

The derivation of $\boldsymbol{h}_{\text{dep}}$ is as follows. First, we use an off-the-shelf dependency parser to get the dependencies of the input text $x = \langle\text{cls}\rangle\ w_1\ w_2\ \cdots\ w_n$, then convert each dependency into a flatten text, obtaining $\text{DepText} = \{\text{DT}_1, \text{DT}_2, \ldots, \text{DT}_n\}$. After that, we let these texts each go through into a BERT module, receiving their vector representations (i.e., $\boldsymbol{h}_1^{\text{dep}}\boldsymbol{h}_2^{\text{dep}} \cdots \boldsymbol{h}_n^{\text{dep}}$) by the corresponding $\langle\text{cls}\rangle$ symbol:

$$\boldsymbol{h}_i^{\text{dep}} = BERT^R(\ \text{DT}_i\ ). \tag{1}$$

Then, we exploit another BERT to compute the text-level representation of the original input sentence:

$$\boldsymbol{h}_{\langle\text{cls}\rangle} = BERT^S(\ \langle\text{cls}\rangle\ w_1\ w_2\ \cdots\ w_n\ ), \tag{2}$$

which are used to guide the following attentive value computation of dependency syntax representations. Finally, we use a simple attention neural network to aggregate these representations into a single vectorial summarization:

$$\begin{pmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{pmatrix} = \text{Softmax}\begin{pmatrix}\boldsymbol{h}_1^{\text{dep}}\boldsymbol{T}\boldsymbol{h}_{\langle\text{cls}\rangle} + b\\ \vdots\\ \boldsymbol{h}_n^{\text{dep}}\boldsymbol{T}\boldsymbol{h}_{\langle\text{cls}\rangle} + b\end{pmatrix} \tag{3}$$
$$\boldsymbol{h}_{\text{dep}} = \sum_{i=1}^{n}\alpha_i\boldsymbol{h}_i^{\text{dep}},$$

where $\boldsymbol{T}$ and b are tunable parameters in the attention module. Particularly, we tune the BERT model $BERT^R$ in retriever component while freeze the BERT model $BERT^S$ in the BERT-BiLSTM-CRF OEI component during training, aiming to strengthen the importance of syntax information.

As a result of these attentive values $\alpha_1\alpha_2\cdots\alpha_n$ by the BERT-based opinion expression identification model, we can also filter the corresponding dependency descriptions by a hyperparameter τ synchronously. The kept dependency descriptions $\{\text{DT}_i\}$ are joined sequentially by the order of α_is, resulting in the final description of dependency syntax, which is then incorporated into the prompt text to support LLM reasoning for OEI. Concretely, we rewrite {Task Description} and {Input} in the prompt of the LLM-based baseline OEI model to tell the large language model to refer to extra dependency syntax information during OEI reasoning. As illustrated in the blue text in Fig. 2, we add the sentence "You can refer to the supported dependency syntax information, which might help you to solve the task". into {Task Description}. For {Input}, we insert selected dependency descriptions (i.e., and the key dependency syntax: the syntactic head of "great" is "was" with the relation adjectival complement, the syntactic head of "long" is "drive" with the relation adjectival modifier) into the original prompt.

4. Experiments

We conduct experiments to verify the effectiveness of our proposed LLM-based OEI model, and also perform detailed analyses to gain a deeper understanding of our approach.

4.1. Experimental settings

Datasets. We conduct experiments on three datasets: English MPQA v2.0 [1], Chinese CH-SIMS [49] and Portuguese MPQA datasets. For the English MPQA, we adopt the same data split as Zhang et al. [18] and Wu et al. [40] for evaluation. The original CH-SIMS does not include answers of opinion expressions. Here we supplement it with OEI annotations manually which may also facilitate future multi-modal opinion mining. For the Portuguese MPQA dataset, it is constructed by Almeida et al. [5] by translation and aligning. Table 1 shows the detailed dataset statistics.

Table 1
Dataset statistics. #Sent and #Exp are the number of sentences and expressions, respectively. AvgSL and AvgEL are the average length of sentences and expressions, respectively.

Dataset	Split	#Sent	#Exp	AvgSL	AvgEL
EN-MPQA	train	4495	1630	24.89	2.37
	dev	1620	543	23.20	2.12
	test	1680	478	24.26	2.03
CH-SIMS	train	1215	1818	15.76	5.24
	dev	151	192	12.85	5.17
	test	151	186	13.30	5.88
PT-MPQA	train	1234	600	33.54	6.31
	dev	154	80	32.51	7.03
	test	154	89	34.92	5.80

Table 2
Main results.

Model	Size	Baseline			+Syntax			Δ_F
		P	R	F	P	R	F	
EN-MPQA								
BERT	110M	42.7	40.6	41.4	**45.4**	**42.3**	**43.9**	+2.5
LLAMA3	8B	45.0	43.1	44.1	**46.5**	**44.3**	**45.4**	+1.3
LLAMA3	70B	50.2	48.3	49.2	**51.5**	**49.5**	**50.5**	+1.3
QWen2	7B	43.3	41.1	42.2	**46.1**	**43.2**	**44.6**	+2.4
QWen2	72B	48.6	46.8	47.7	**49.7**	**47.9**	**48.8**	+1.1
CH-SIMS								
BERT	110M	43.1	41.0	42.0	**44.6**	**41.8**	**43.1**	+1.1
QWen2	7B	43.9	41.8	42.8	**45.1**	**42.5**	**43.8**	+1.0
QWen2	72B	48.4	46.8	47.6	**50.0**	**48.1**	**49.0**	+1.4
PT-MPQA								
BERT	110M	32.5	30.1	31.3	**33.7**	**31.2**	**32.4**	+1.1
LLAMA3	8B	33.4	31.3	32.3	**34.9**	**32.5**	**33.7**	+1.4
LLAMA3	70B	40.0	37.2	38.5	**41.6**	**38.4**	**39.9**	+1.4

Table 3
Comparison with previous methods.

Model	EN-MPQA	CH-SIMS	PT-MPQA
Decoder-based models			
Jia et al. [14]$_{\text{baseline}}$	49.2	47.6	38.5
This work	**50.5**	**49.0**	**39.9**
Encoder-Decoder models			
Barnes et al. [23]	47.2	45.1	34.7
Samuel et al. [31]	46.2	45.0	34.4
Zhai et al. [24]	47.4	45.5	35.0
Zhou et al. [25]	**47.9**	**46.1**	**35.5**

Hyperparameters. For BERT-based models, we employ bert-base-uncased, bert-base-chinese and BERTimbau-Base [50] to conduct English, Chinese and Portuguese experiments, respectively. For large language model, we fine-tune LLAMA3-instruct (8B/70B) [51], Qwen2-Instruct (7B/72B) [52] with the LORA technique. For model training, we use the AdamW algorithm with learning rate 1e−5, batch size 32, weight decay 0.01 and linear learning rate warmup over the first 400 steps to optimize parameters. We train an off-the-shelf dependency parser [45] based on the PTB [53], CTB [54] and Porttinari-base [55] datasets to parse dependency trees. The threshold of τ is set by 0.6 to filter the redundant dependencies.
Evaluation. We take the span-based precision (**P**), recall (**R**) and F1 score (**F**) as evaluation metrics. Each model is run by 5 times where the median results are shown.

4.2. Main results

Table 2 shows the main results of our investigated models. For the English dataset, we compare the BERT, LLAMA3 and Qwen2-based

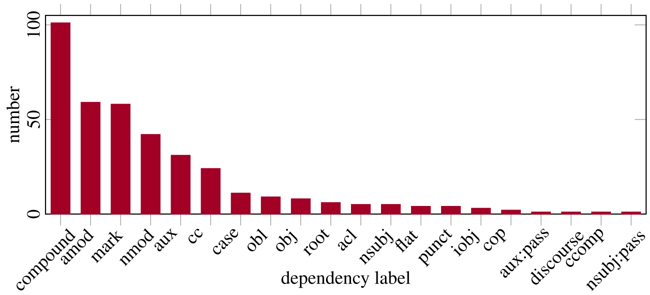

Fig. 3. Syntactic label distribution of the selected dependencies.

models. We can see that large language models can bring significantly better results on the dataset, with an improvement of $49.2 - 41.4 = 7.8$ on the F-value from the BERT-based model to the LLAMA3-70B one. As the model size increases, we can expect larger gains from large language models. Furthermore, our approach with dependency syntax integration is able to yield more gains with significant improvements (The p-value is below 10^{-5} under pairwise t-test). The results are consistent across all investigated models. This observation demonstrates that dependency syntax can be beneficial for LLM-based OEI models as well.

For the Chinese and Portuguese datasets, the tendency is exactly the same. First, by using Qwen2-72B or LLAMA3-70B, we can obtain much better performance than the BERT-based CRF model. Second, the dependency syntax can further boost the OEI performance greatly. We also investigate the OEI performance by using close-sourced GPT models for both datasets. The results are very bad (>5 points below the BERT-based model), where the reason behind might be due to the unchangeable parameters, which indicates that instruct-tuning is very important for the OEI task.

Further, we compare our method with previous state-of-the-art systems. For the LLM-based OEI, the only previous system is our baseline [14]. For traditional encoder–decoder models, we introduce four previous models, which all exploit BERT as encoder backbone. [23,25,31] are all graph-based models, while Zhai et al. [24] exploits a table-filing framework. Table 3 shows the results. Particularly we reimplement or use their codes for fair comparisons, since the benchmark datasets in their original paper are almost mutually different. As shown, Zhou et al. [25] obtains the best-performing results, which is still below our baseline (EN-MPQA and PT-MPQA using LaMMA3-70B, CH-SIMS using QWen2-72B). As whole, we find that the LLM-based methods lead to better results on our OEI datasets when the scale of base model is large enough.

4.3. Analyses

We also conduct several analysis work on the English MPQA dataset here in order to understand our opinion expression identification model in-depth.

4.3.1. Dependency label distribution

First, we examine which types of syntactic dependencies are chosen for our opinion expression identification task. There are many different syntactic labels in dependency syntax. Each dependency label implies a specific syntactic and semantic relation between the head word and the dependent word. Hence, dependency labels with diverse syntactic functions might contribute to opinion expression identification in various ways. Intuitively, the syntactic labels which are closely associated with opinion expressions should be captured. Fig. 3 shows the dependency label distribution of the selected dependencies in the development set, where the hit frequencies are reported. As shown, taking the top-2 labels "compound" and "amod (adjectival complement)" as example, the first label can help to ensure the completeness of opinion expressions, e.g., "terrorist groups" and "holiday season", while the second label can help to locate opinion expressions, e.g., "great/terrible" and "happy/sad". As a whole, we find that all high-frequency syntactic labels contribute directly to the OEI task, which is consistent with our expectations.

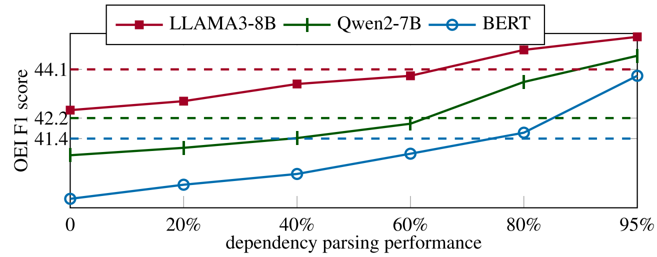

Fig. 4. The OEI performance in terms of dependency parsing accuracy.

Table 4
The averaged number of kept dependency arcs.

Threshold	0.0	0.2	0.4	0.6	0.8	1.0
Dependency number	24.73	1.25	1.05	0.98	0.97	0.00
F1 score	43.6	44.8	45.0	**45.4**	45.1	44.1

4.3.2. Dependency parsing performance

Then, we check how dependency parsing performance affects opinion expression identification. Generally speaking, the performance of the syntax parser will influence the precision of the task of the downstream application by syntactic trees with varying degrees of accuracy. Concretely, we train several dependency parsers by using training datasets of different sizes, controlling the dependency accuracy by around 20%, 40%, 60%, 80% and 95% (the best performance) on a standard PTB test dataset. After that, we perform our syntax-enhanced LLM-based OEI model and evaluate the opinion expression identification performance. Fig. 4 shows the OEI performance in terms of dependency parsing accuracy. We can see that the OEI performance is boosted as the parsing performance increases across all investigated models, including BERT, QWen2-7B and LLAMA3-8B. The results indicate that dependency syntax is positively related to OEI.

4.3.3. Filtering threshold and dependency number

Thirdly, we show how many dependencies are finally selected to support OEI by our method, and meanwhile study the sensitivity of our filtering threshold τ since the dependency selection is controlled by τ. Table 4 shows the results with different thresholds based on LLAMA3-instruct (8B). As shown, the redundant dependencies indeed hurt the LLM reasoning, leading to decreased performance. When the threshold is 0.0, all dependencies are kept and inserted into the prompt. F1 score of opinion expression identification is 43.6, indicating the whole dependency tree harms OEI performance. In addition, the number of selected dependencies in our final model is only around 1, which greatly reduces the length of the prompt.

4.3.4. Computational cost of key components

Finally, we show the computational cost of our model of different components. To fully run our system, we have three steps mainly: (1) dependency parsing, (2) dependency retrieval and (3) LLM reasoning. Here we focus on the test phase. Figure shows the time consumed of the three steps on the whole EN-MPQA test dataset, respectively. Our computational environment is Linux OS on with eight NVIDIA A100-40G GPUs, and the select base LLMs are LLAMA3-instruct 8B and 70B. As shown in Table 5, we can see that the main cost of our system is at the LLM reasoning, where the other part can be largely ignorable. Evening for the LLAMA3-instruct 8B system, the extra computation cost of dependency parsing and dependency retrieval only occupies less then 1%.

Table 5
The speed comparison of key components of our method.

LLAMA3-instruct	8B		70B
Dependency parsing		31 s	
Dependency retrieval		88 s	
LLM reasoning	4.3 h		6.1 h

5. Discussion and implication

This study helps us understand the function of dependency syntax for LLM-based OEI. In the below, we show the main findings in the term of theoretical and practical contributions.

5.1. Theoretical contribution

The theoretical implications of this work are two folds. First, this work provides one feasible strategy to integrate dependency syntax for the LLM-based paradigm. Although dependency syntax has been widely-adopted in previous encoder–decoder methods, the line of research is very seldom for decoder-only LLM-based models. Without encoder, it is difficult to represent structural-formed syntax information. Our work presents a solution and can be easily extendable to other tasks. Second, from the view of OEI, we further show that knowledgable linguistic information is closely related with the OEI task. Although LLMs have been demonstrated strong capability of semantic understanding, they are still complementary with the knowledge designed by linguistic experts. Dependency syntax is one of this kind, and we may also benefit from others, e.g., abstract meaning representation [56].

5.2. Practical contribution

The practical implications of our work lie in the task of OEI, which is a core task of opinion mining. Our proposed method achieves a new state-of-the-art performance in OEI, thereby enhancing the effectiveness of various opinion mining applications. For example, our system can help for product improvement and pricing, online advertising, competitive intelligence, opinion Monitoring and etc. Although our method requires huge support of computing power, we can exploit strategies such as knowledge distilling to strength small-scale encoder–decoder models [57].

6. Conclusion and future works

In this work, we presented the first work of LLM-based OEI with dependency syntax. The key idea is to augment the prompt text of a baseline OEI with the respective text reflecting the dependency structures. We addressed two main issues of the integration. First, we convert the structural-formed dependency syntax into a collection of flat texts, in order to align with prompt learning of LLM-based OEI. Concretely, a full dependency parsing tree is transformed into a collection of textual descriptions by decomposing it into dependencies. Second, we join the collection of texts into a single text and meanwhile eliminate the redundant parts by a BERT-based CRF model as a dependency retriever, enabling LLM with effective reasoning. Specifically, we exploited a BERT-based CRF model to weight the collection of dependency descriptions, keeping only salient dependencies for OEI. Our experiments show that the proposed method is highly effective with significant improvements on two benchmark datasets. Furthermore, the integration of dependency syntax could be applicable to many other NLP tasks, providing a promising direction for future research.

There are also several limitations of this work, which can be addressed in future works. First, we exploit a template-based method to convert dependency trees into texts, and the inside connections between individual dependencies are not considered. Second, we only evaluate the methods on the English, Chinese and Portuguese datasets, and actually the really low-resource and morphologically-rich languages are not comprehensively investigated. Additionally, multi -lingual and cross-lingual conditions should also be carefully examined. Third, our method consumes computational resources heavily, which could be a serious issue in the real-world setting, particularly for the real-time scenario. We may use teacher-student knowledge distilling techniques to solve the problem. Finally, under the LLM paradigm, the zero-shot and few-shot settings are more practical, while this may be accomplished in the future by detailing the infer steps with techniques such as chain-of-thought and slow thinking.

CRediT authorship contribution statement

Qiujing Xu: Writing – review & editing, Writing – original draft, Software, Methodology. **Peiming Guo:** Writing – original draft, Software, Methodology. **Fei Li:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Meishan Zhang:** Writing – review & editing, Supervision, Methodology, Investigation. **Donghong Ji:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) (No. 62176187).

Data availability

Data will be made available on request.

References

[1] J. Wiebe, T. Wilson, C. Cardie, Annotating expressions of opinions and emotions in language, Lang. Resour. Eval. 39 (2005) 165–210.
[2] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 2 (1–2) (2008) 1–135.
[3] B. Liu, Sentiment analysis and opinion mining, Synthesis Lectures on Human Language Technologies, 2012.
[4] O. Irsoy, C. Cardie, Opinion mining with deep recurrent neural networks, in: Proceedings of EMNLP 2014, 2014, pp. 720–728.
[5] M.S. Almeida, C. Pinto, H. Figueira, P. Mendes, A.F. Martins, Aligning opinions: Cross-lingual opinion mining with dependencies, in: Proceedings of ACL, 2015, pp. 408–418.
[6] M. Zhang, Q. Wang, G. Fu, End-to-end neural opinion extraction with a transition-based model, Inf. Syst. 80 (2019) 56–63.
[7] R. Bar-Haim, E. Dinur, R. Feldman, M. Fresko, G. Goldstein, Identifying and following expert investors in stock microblogs, in: Proceedings of the EMNLP, 2011, pp. 1310–1319.
[8] H. Rui, Y. Liu, A. Whinston, Whose and what chatter matters? The effect of tweets on movie sales, Decis. Support Syst. 55 (4) (2013) 863–870.
[9] M. Sidorov, W. Minker, Emotion recognition and depression diagnosis by acoustic and visual features: A multimodal approach, in: Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge, AVEC '14, New York, NY, USA, 2014, pp. 81–86.
[10] Y. Ma, Y. Zhang, M. Bachinski, M. Fjeld, Emotion-aware voice assistants: Design, implementation, and preliminary insights, in: Proceedings of the Eleventh International Symposium of Chinese CHI, 2023, pp. 527–532.
[11] E. Breck, Y. Choi, C. Cardie, Identifying expressions of opinion in context, in: Proceedings of IJCAI, 2007, pp. 2683–2688.
[12] P. Liu, S. Joty, H. Meng, Fine-grained opinion mining with recurrent neural networks and word embeddings, in: Proceedings of the EMNLP, Lisbon, Portugal, 2015, pp. 1433–1443.
[13] A. Katiyar, C. Cardie, Investigating LSTMs for joint extraction of opinion entities and relations, in: Proceedings of the 54th ACL, Berlin, Germany, 2016, pp. 919–929.

[14] B. Jia, H. Chen, Y. Sun, M. Zhang, M. Zhang, LLM-Driven multimodal opinion expression identification, in: Proceedings of InterSpeech, 2024.

[15] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, 2023, arXiv preprint arXiv:2303.18223.

[16] W. Zhang, Y. Deng, B. Liu, S. Pan, L. Bing, Sentiment analysis in the era of large language models: A reality check, in: Findings of the NAACL, 2024, pp. 3881–3906.

[17] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, ACM Trans. Intell. Syst. Technol. 15 (3) (2024).

[18] B. Zhang, Y. Zhang, R. Wang, Z. Li, M. Zhang, Syntax-aware opinion role labeling with dependency graph convolutional networks, in: Proceedings of ACL, 2020, pp. 3249–3258.

[19] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Adv. Neural Inf. Process. Syst. 33 (2020) 9459–9474.

[20] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2023, arXiv preprint arXiv:2312.10997 2.

[21] M. Hu, B. Liu, Mining opinion features in customer reviews, in: AAAI, Vol. 4, 2004, pp. 755–760.

[22] E. Cambria, B. Schuller, Y. Xia, C. Havasi, New avenues in opinion mining and sentiment analysis, IEEE Intell. Syst. 28 (2) (2013) 15–21.

[23] J. Barnes, R. Kurtz, S. Oepen, L. Øvrelid, E. Velldal, Structured sentiment analysis as dependency graph parsing, in: Proceedings of ACL, 2021, pp. 3387–3402.

[24] Z. Zhai, H. Chen, R. Li, X. Wang, USSA: A unified table filling scheme for structured sentiment analysis, in: Proceedings of ACL, 2023, pp. 14340–14353.

[25] C. Zhou, B. Li, H. Fei, F. Li, C. Teng, D. Ji, Revisiting structured sentiment analysis as latent dependency graph parsing, in: Proceedings of ACL, 2024, pp. 10178–10191.

[26] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, S. Patwardhan, OpinionFinder: A system for subjectivity analysis, in: Proceedings of EMNLP 2005 Interactive Demonstrations, 2005, pp. 34–35.

[27] Y. Choi, E. Breck, C. Cardie, Joint extraction of entities and relations for opinion recognition, in: Proceedings of the EMNLP, 2006, pp. 431–439.

[28] B. Yang, C. Cardie, Extracting opinion expressions with semi-markov conditional random fields, in: Proceedings of EMNLP, 2012, pp. 1335–1345.

[29] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, B. Xu, Joint extraction of entities and relations based on a novel tagging scheme, in: Proceedings of the 55th ACL, 2017, pp. 1227–1236.

[30] X. Zhang, G. Xu, Y. Sun, M. Zhang, X. Wang, M. Zhang, Identifying Chinese opinion expressions with extremely-noisy crowdsourcing annotations, in: Proceedings of ACL, 2022, pp. 2801–2813.

[31] D. Samuel, J. Barnes, R. Kurtz, S. Oepen, L. Øvrelid, E. Velldal, Direct parsing to sentiment graphs, in: Proceedings of the 60th ACL (Short Papers), 2022, pp. 470–478.

[32] A. Albladi, M. Islam, A. Das, M. Bigonah, Z. Zhang, F. Jamshidi, M. Rahgouy, N. Raychawdhary, D. Marghitu, C. Seals, Hate speech detection using large language models: A comprehensive review, IEEE Access (2025).

[33] A. Culotta, J. Sorensen, Dependency tree kernels for relation extraction, in: Proceedings of the 42nd ACL, 2004, pp. 423–429.

[34] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, in: Proceedings of ACL-IJCNLP, 2015, pp. 1556–1566.

[35] Z. Xu, D. Guo, D. Tang, Q. Su, L. Shou, M. Gong, W. Zhong, X. Quan, D. Jiang, N. Duan, Syntax-enhanced pre-trained model, in: Proceedings of the 59th ACL-IJCNLP, 2021, pp. 5412–5422.

[36] J. Bai, Y. Wang, Y. Chen, Y. Yang, J. Bai, J. Yu, Y. Tong, Syntax-BERT: Improving pre-trained transformers with syntax trees, in: Proceedings of the 16th EACL, 2021, pp. 3011–3020.

[37] M. Joshi, C. Rosé, Generalizing dependency features for opinion mining, in: Proceedings of the ACL-IJCNLP (Short Papers), 2009, pp. 313–316.

[38] X. Zhu, P. Sobihani, H. Guo, Long short-term memory over recursive structures, in: International Conference on Machine Learning, PMLR, 2015, pp. 1604–1612.

[39] Q. Xia, B. Zhang, R. Wang, Z. Li, M. Zhang, A unified span-based approach for opinion mining with syntactic constituents, in: Proceedings of NAACL, 2021, pp. 1795–1804.

[40] S. Wu, H. Fei, F. Li, M. Zhang, Y. Liu, C. Teng, D. Ji, Mastering the explicit opinion-role interaction: Syntax-aided neural transition system for unified opinion role labeling, Proc. AAAI 36 (10) (2022) 11513–11521.

[41] O. Levy, Y. Goldberg, Dependency-based word embeddings, in: Proceedings of the 52nd ACL (Short Papers), 2014, pp. 302–308.

[42] M. Zhang, Z. Li, G. Fu, M. Zhang, Dependency-based syntax-aware word representations, Artificial Intelligence 292 (2021) 103427.

[43] S. Kübler, R. McDonald, J. Nivre, Dependency parsing, in: Dependency Parsing, Springer, 2009, pp. 11–20.

[44] T. Dozat, C.D. Manning, Deep biaffine attention for neural dependency parsing, in: International Conference on Learning Representations, 2017.

[45] Y. Li, Z. Li, M. Zhang, R. Wang, S. Li, L. Si, Self-attentive biaffine dependency parsing, in: IJCAI, 2019, pp. 5067–5073.

[46] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, J. Li, A survey of knowledge enhanced pre-trained language models, IEEE Trans. Knowl. Data Eng. 36 (4) (2023) 1413–1430.

[47] L. Yang, H. Chen, Z. Li, X. Ding, X. Wu, Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling, IEEE Trans. Knowl. Data Eng. 36 (7) (2024) 3091–3110.

[48] W. Zhong, L. Guo, Q. Gao, H. Ye, Y. Wang, Memorybank: Enhancing large language models with long-term memory, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 19724–19731.

[49] W. Yu, H. Xu, F. Meng, Y. Zhu, Y. Ma, J. Wu, J. Zou, K. Yang, CH-SIMS: A Chinese multimodal sentiment analysis dataset with fine-grained annotation of modality, in: Proceedings of ACL, 2020, pp. 3718–3727.

[50] F. Souza, R. Nogueira, R. Lotufo, Bertimbau: Pretrained BERT models for Brazilian portuguese, in: R. Cerri, R.C. Prati (Eds.), Intelligent Systems, 2020, pp. 403–417.

[51] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, 2023, arXiv preprint arXiv:2307.09288.

[52] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, et al., Qwen2 technical report, 2024, arXiv preprint arXiv:2407.10671.

[53] M.P. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank, Comput. Linguist. 19 (2) (1993) 313–330.

[54] N. Xue, F. Xia, F.-D. Chiou, M. Palmer, The penn chinese treebank: Phrase structure annotation of a large corpus, Nat. Lang. Eng. 11 (2) (2005) 207–238.

[55] M.S. Duran, L. Lopes, M.d.G.V. Nunes, T.A.S. Pardo, The dawn of the porttinari multigenre treebank: Introducing its journalistic portion, in: Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana, STIL, SBC, 2023, pp. 115–124.

[56] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider, Abstract meaning representation for sembanking, in: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, 2013, pp. 178–186.

[57] J. Gou, B. Yu, S.J. Maybank, D. Tao, Knowledge distillation: A survey, Int. J. Comput. Vis. 129 (6) (2021) 1789–1819.