

Enhancing Language Representation with Constructional Information for Natural Language Understanding

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Zhilin Gong, Ming Cai*, Tianxiang Wang

Department of Computer Science and Technology, Zhejiang University

{xlxw, wujw, pengjw, zhilingong, cm, wang_tx}@zju.edu.cn

Abstract

Natural language understanding (NLU) is an essential branch of natural language processing, which relies on representations generated by pre-trained language models (PLMs). However, PLMs primarily focus on acquiring lexico-semantic information, while they may be unable to adequately handle the meaning of constructions. To address this issue, we introduce construction grammar (CxG), which highlights the pairings of form and meaning, to enrich language representation. We adopt usage-based construction grammar as the basis of our work, which is highly compatible with statistical models such as PLMs. Then a HyCxG framework is proposed to enhance language representation through a three-stage solution. First, all constructions are extracted from sentences via a slot-constraints approach. As constructions can overlap with each other, bringing redundancy and imbalance, we formulate the conditional max coverage problem for selecting the discriminative constructions. Finally, we propose a relational hypergraph attention network to acquire representation from constructional information by capturing high-order word interactions among constructions. Extensive experiments demonstrate the superiority of the proposed model on a variety of NLU tasks.

1 Introduction

Recent progress in natural language processing relies on pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020; Brown et al., 2020; Raffel et al., 2020; He et al., 2021), which generate word representations by capturing syntactic and semantic features from their context. Typically, PLMs (e.g., BERT) employ the Masked Language Model (MLM) as a pre-training objective, randomly masking some tokens from the input before predicting the original tokens. However, PLMs primarily focus on acquiring lexico-semantic information (Tayyar Madabushi et al.,

* Corresponding author.

Sentence	Construction
The [staff] _N should be a bit more friendly.	NOUN–AUX–be (staff–should–be)
The restaurants try too hard to make fancy [food] _N .	ADV–hard–to (too–hard–to)
I can understand the prices if it served better [food] _N .	if–PRON–VERB (if–it–served)

Table 1: Visualization of attention scores from misclassified examples in aspect-based sentiment analysis. [·]_N refers to the aspects with negative sentiment polarity.

2020) while the meaning is not attached to the words instantiating the construction, but rather to the abstract pattern itself (Weissweiler et al., 2023). Learning the representation of more complex and abstract linguistic units (called constructions) can be more challenging, such as collocations, argument structures and sentence patterns. For example, the ditransitive construction (Goldberg, 1995), which involves the form “Subject–Verb–Object1–Object2”, denotes the meaning of transfer or giving of an entity (Object2) from an agent (Subject) to a recipient (Object1), such as “John gave Mary a book”. Linguistic experiments have proved that constructions can substantially contribute to sentence comprehension in multiple languages, such as English (Bencini and Goldberg, 2000) and German (Gries and Wulff, 2005).

As the meaning of a construction is assigned to a language pattern rather than specific words (Hoffmann and Trousdale, 2013), the constructional information is rarely acquired by MLM and requires large bulk training data, which may lead to failure in natural language understanding tasks with constrained low resource data. As shown in Table 1, we illustrate three examples from aspect-based sentiment analysis (ABSA) tasks. The attention scores are derived from BERT-SPC (Devlin et al., 2019), a simple baseline by fine-tuning

aspect-specific representation of BERT. In the first example, the model focuses mainly on the opinion word “friendly” while ignoring the modality construction of “should be”, resulting in the misclassification. In the second case, “too hard to” expresses a negation for the “fancy food”. However, the model ignores the collocation of “too...to” structure, and the “food” is wrongly considered to be positive. The third example illustrates that lack of understanding of conditional sentence (*if* clause) causes the model to make an incorrect prediction.

Observations from both linguistics and NLU standpoints motivate us to exploit construction grammar (CxG; Goldberg, 1995, 2006) as the inductive bias to complement language representation with constructional information. Construction grammar refers to a family of linguistic approaches which regard constructions as the basic unit of language. Composed of pairings of form and meaning, constructions pervade all levels of language along a continuum (e.g., morphology, lexicon and syntax).

However, when bridging the gap between CxG and NLU, we face three critical questions:

- (Q1) Which CxG approach is applicable for NLU?
- (Q2) How can typical constructions be identified?
- (Q3) How can constructions be encoded?

To answer (Q1), we investigate different variants of CxG. Instead of formal methods (e.g., fluid construction grammar and embodied construction grammar), we adopt usage-based approaches which assume that grammar acquisition involves statistical inference. Therefore constructions can be induced as frequent instances of linguistic units. This assumption makes usage-based approaches highly compatible with statistical models such as PLMs (Kapatsinski, 2014). Specifically, we follow the efforts of Dunn (2019) with a computationally slot-constraints approach that formulates constructions as the combinations of immutable components and fillable slots. As shown in Table 1, the modality construction “NOUN–AUX–*be*” in the first example contains an immutable component “*be*” and two fillable slots (NOUN and AUX) with a meaning of expressing advice or suggestion.

As for (Q2), from the usage-based perspective, constructions are often stored redundantly at different levels of language. Therefore constructions can overlap with each other, which results in redundancy and imbalance. Consequently, we formulate the problem of selecting discriminative constructions as multi-objective optimization for con-

ditional maximum coverage. To alleviate the computational complexity, we adopt a heuristic search algorithm, i.e., simulated annealing (SA) to determine the composition of constructions.

In order to address (Q3), we propose a relational hypergraph attention network (R-HGAT) to capture high-order word interactions inside the constructions, so as to acquire representation from constructional information. R-HGAT generalizes hypergraph attention network using its flexible labeled hyperedges. We refer to the entire framework which involves construction extraction, selection and encoding as hypergraph network of construction grammar (HyCxG)¹.

Extensive experiments have been conducted to illustrate the superiority of HyCxG on NLU tasks, while multilingual experiments further indicate the constructional information is beneficial across languages. Additionally, based on the constructional representations acquired by our model, we conduct an empirical study of building a network of constructions, which provides meaningful implication and in turn contributes to usage-based CxG.

2 Construction Extraction and Selection

In this section, we elaborate on the details of extraction (Q1) and selection of constructions (Q2).

2.1 Computational Construction Grammar

Construction grammar is a branch of cognitive linguistics which assumes that syntax and patterns have specific meanings. A construction is a form-meaning pair whose structure varies in different level of abstractness, including partially or fully filled components and general linguistic patterns. According to CxG, speakers can recognize patterns after coming across them a certain number of times, comparable to merging n-grams (i.e., constructions) at different schematic or abstract levels (Goldberg, 2003, 2006; Tayyar Madabushi et al., 2020). However, the methodology for acquiring the constructions is always labor-intensive and requires careful definition, while the computational generation of CxG is a relatively new research field.

As our answer to (Q1), the usage-based approach is employed to obtain constructions following the work of Dunn (2019). They propose a grammar induction algorithm, which can learn the patterns of constructions with a slot-constraints approach from

¹Our code is publicly available at <https://github.com/xlwalex/HyCxG>.

the corpus. Therefore, we adapt [Dunn’s \(2019\)](#) system in our work for construction extraction.

Constructions are represented as a combination of slots and are separated by dashes (Table 1). The slot fillers are drawn from lexical units (words), semantic categories (the discrete semantic domains are clustered with fastText embeddings ([Grave et al., 2018](#)) via K-Means) and syntax (Universal Part-of-Speech tags; [Petrov et al., 2012](#)), which are progressively higher in abstraction level.

2.2 Conditional Max Coverage Problem

Since constructions are often stored redundantly at different levels of abstractness, overlapping constructions can be captured by the grammar induction algorithm ([Dunn, 2017, 2019](#)). We summarize the phenomenon of overlap into two scenarios: *Inclusion* and *Intersection*. *Inclusion* refers to the case where one construction is a subpart or instantiation of another construction, while *Intersection* indicates that the constructions have some common slots. Redundancy and imbalance encoding problem can be brought by overlap phenomenon. The redundancy of the constructions can cause high computational complexity. And imbalanced construction distribution may introduce interference. This is due to the fact that the words in high dense parts are updated frequently, while this seldom happens in low dense parts. There is a virtual example to illustrate these phenomena in Figure 1.

Therefore, the key issue is to select the discriminative constructions from all candidates (Q2). We formulate this as multi-objective optimization for conditional maximum coverage (Cond-MC) problem. Specifically, we seek an optimal subset $\mathcal{C}_O = \{c_1, c_2, \dots, c_m\}$ from the set $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ containing all n constructions. The constructions in \mathcal{C}_O are supposed to reach the following three objectives:

1. The constructions cover as many words as possible in sentences.
2. Intersection among constructions is minimal.
3. Constructions preferably contain more slots from concrete levels.

Objectives 1 and 2 spread the constructions throughout the entire sentence to ensure balanced distribution, while the more discriminative constructions can be selected by objective 3. To unify the objective functions for optimization, we formalize each objective as a specific score function and weight them together to generate the total score. Therefore, Cond-MC can be converted

Algorithm 1: SA algorithm for Cond-MC

Input: The set of all constructions \mathcal{C} .

Output: The optimal set \mathcal{C}_O .

```

1 Initialize the feasible solutions  $\mathcal{C}_O \leftarrow \mathcal{C}_f$ 
2 Initialize temperature  $T \leftarrow T_0$ , step  $k \leftarrow 0$ 
3 while  $k < k_{max}$  do
4    $T \leftarrow \text{COOL}(T, k, k_{max})$ 
5    $\mathcal{C}_k \leftarrow \text{RANDFLIP}(\mathcal{C}_O, T)$ 
6    $dE \leftarrow \text{SCORE}(\mathcal{C}_k) - \text{SCORE}(\mathcal{C}_O)$ 
7   if  $dE > 0$  or  $\text{RAND}() \leq P(dE, T)$  then
8      $\mathcal{C}_O \leftarrow \mathcal{C}_k$ 
9   end
10   $k \leftarrow k + 1$ 
11 end
12 return  $\mathcal{C}_O$ 

```

to the problem of maximizing the score for \mathcal{C}_O . We define each construction c_i as a set of r slots $c_i = \{w_1, w_2, \dots, w_r\}$, while $D(\cdot)$ is utilized to compute the number of elements in a set. Then the score function of objectives 1 and 2 can be formulated as $D(c_1 \cup \dots \cup c_m)$ and $D(c_1 \cap \dots \cap c_m)$, with S_{ob1} and S_{ob2} referring to them respectively. And the score function of objective 3 is written as:

$$S_{ob3} = \sum_{i=1}^m \sum_{j=1}^r \frac{s_r}{r}, s_r \in \{s_{\text{syn}}, s_{\text{sem}}, s_{\text{lex}}\} \quad (1)$$

where three scores s_{syn} , s_{sem} and s_{lex} are assigned for slots in syntax, semantic and lexical level. The score of each construction is calculated by averaging the slot scores, while S_{ob3} computes the total score of all constructions in \mathcal{C}_O . Then Cond-MC can be formulated as:

$$\begin{cases} \max & w_1 S_{ob1} - w_2 S_{ob2} + w_3 S_{ob3} \\ \text{s.t.} & \mathcal{C}_O \subset \mathcal{C} \end{cases} \quad (2)$$

where w_1 , w_2 and w_3 are the weighting factors for balancing the scores of three objectives.

2.3 Solution for Cond-MC Problem

As we formulate a multi-objective optimization problem for selecting typical constructions, it is complicated to solve Cond-MC since it is an NP problem. When the solution space is large, i.e., there is a significant amount of constructions in the sentence, which leads to an unacceptable computational complexity. To alleviate this issue, we employ the Simulated Annealing (SA; [Kirkpatrick et al., 1983](#)) algorithm in heuristic search of \mathcal{C}_O , where the procedure is stated in Algorithm 1.

We define \mathcal{C}_O as a binary set to indicate whether a construction is selected. $\text{RANDFLIP}(\cdot)$ is applied to reverse t elements stochastically, while step k is inversely related to temperature T . $\text{SCORE}(\cdot)$ is the overall score function in Equation 2. Besides, we utilize $\text{RAND}()$ to compare with the transition probability P to estimate whether to accept the new solution under metropolis criterion (i.e., accepting a new solution with a certain probability to increase the perturbation). A cooling schedule is used to control the evolution of the new solution with a reduced temperature function $\text{COOL}(\cdot)$ to regulate the temperature in an exponential form.

3 Model

After extracting and determining the optimal set for constructions, the next challenge is to encode constructions in an effective way (Q3). In this section, we first introduce hypergraph for modelling complex data correlation within constructions. Then a relational hypergraph attention network (R-HGAT) is proposed to encode the high-order word interactions inside each construction. Finally, enhanced by the constructional information, the language representation is ready for NLU tasks.

3.1 Construction Hypergraph Generation

Since the associations within constructions are not only dyadic, but could be triadic, tetradic or even higher-order, we adopt hypergraph for modelling such data correlation. Hypergraph is a generalization of conventional graph that the hyperedge can connect arbitrary number of nodes. A hypergraph can be defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, which contains a node set $\mathcal{V} = \{v_1, v_2, \dots, v_m\}$ and a hyperedge set $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$. As each hyperedge can connect two or more nodes, the hypergraph \mathcal{G} can be denoted by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, with entries being defined as:

$$\mathbf{H}_{ij} = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{if } v_i \notin e_j \end{cases} \quad (3)$$

Given a sentence with m words (i.e., nodes in \mathcal{G}), we can model high-order word interactions through n typical constructions. In specific, each construction is regarded as a hyperedge, while the words contained in the construction are considered as the member nodes of certain hyperedges. Thus, we can generate the hypergraph for each sentence as illustrated in Figure 1.

3.2 Hypergraph Attention Network

Different from the simple graph that nodes are pairwise linked, the information of nodes cannot be directly aggregated from neighboring nodes in hypergraph. Thus, hypergraph attention network (HGATT) is employed to learn the node representations, which separates the computation procedure into two steps, i.e., node aggregation and hyperedge aggregation. It first aggregates the information of nodes for generating the hyperedge representation. Then the information is fused back to the nodes from hyperedges via hyperedge aggregation. In general, the procedure is formulated as follows:

$$\begin{aligned} \mathbf{g}_j^l &\leftarrow \text{AGGR}_{\text{Node}} \left(\left\{ \mathbf{h}_s^{l-1} \mid \forall v_s \in e_j \right\} \right) \\ \mathbf{h}_i^l &\leftarrow \text{AGGR}_{\text{Edge}} \left(\mathbf{h}_i^{l-1}, \left\{ \mathbf{g}_j^l \mid \forall e_j \in \mathcal{E}_i \right\} \right) \end{aligned} \quad (4)$$

where $\text{AGGR}_{\text{Node}}$ and $\text{AGGR}_{\text{Edge}}$ denote the two-step aggregation functions for nodes and hyperedges. \mathbf{h}_i^l and \mathbf{g}_j^l denote the representations of node v_i and hyperedge e_j in l -th layer, while \mathcal{E}_i is the set of hyperedges connected to node v_i .

The HGATT is mainly implemented based on graph attention mechanism (Veličković et al., 2017), such as HyperGAT (Ding et al., 2020a). However, this attention mechanism applies the same weight matrices for different types of information within hyperedges, which inhibits the ability for the models to capture high-order relations in hypergraphs (Fan et al., 2021).

3.3 R-HGAT Network

To tackle the limitation of HGATT and exploit the information of constructions, we propose a relational hypergraph attention network (R-HGAT). The mutual attention mechanism, i.e., node-level attention and edge-level attention, is adopted to aggregate the information from nodes and hyperedges. The entire architecture is described as follows:

Node-level attention. Node-level attention is utilized to encode representation of hyperedges that aggregates information from the nodes. Given the node v_i and the set of hyperedges \mathcal{E}_i that connect to v_i , we first embed each hyperedge (i.e., construction) into vector space \mathbf{z}_j by looking up an embedding matrix $E_c \in \mathbb{R}^{|\mathcal{V}| \times d}$, while $|\mathcal{V}|$ is the size of construction set within a certain language and d refers to the dimension of the vectors. After that, the attention mechanism is applied to compute the importance score for spotlighting the nodes that

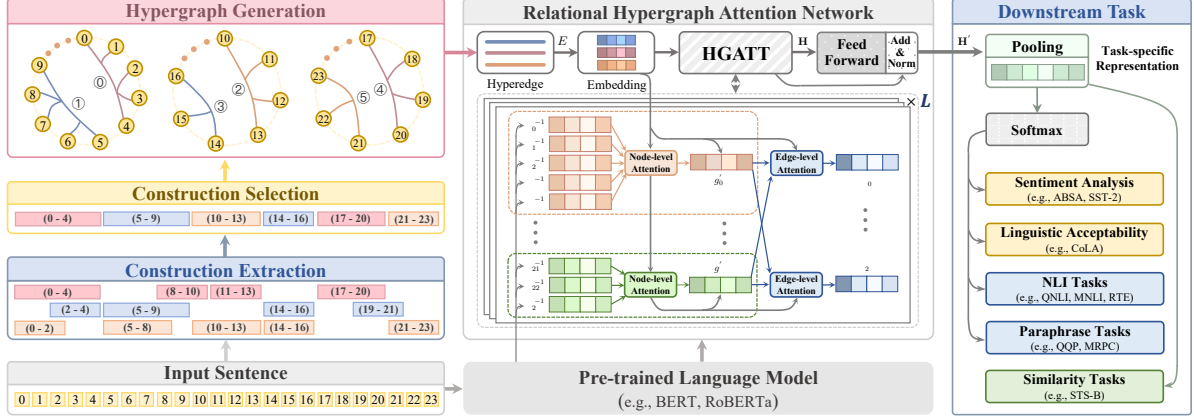


Figure 1: Overview of the proposed HyCxG framework. For illustration, we show the entire system with a virtual example. In construction extraction module, CxG (5-8) is a subpart of CxG (5-9), which demonstrates the *inclusion* relation of the overlap phenomenon, while CxG (5-9) and CxG (8-10) are *intersection* relations.

are crucial to the hyperedge according to \mathbf{z}_j . Then the aggregation procedure can be formulated as:

$$\mathbf{g}_j^l = \sum_{v_s \in e_j} \alpha_{js} W_n \mathbf{h}_s^{l-1} \quad (5)$$

where W_n is the weight matrix and α_{js} is the attention score between the representation of node \mathbf{h}_s^{l-1} and hyperedge \mathbf{z}_j that can be computed by:

$$\mathbf{r}_{js}^{l-1} = \sigma((W_c \mathbf{z}_j)^T W_s \mathbf{h}_s^{l-1})$$

$$\alpha_{js} = \frac{\exp(\mathbf{r}_{js}^{l-1})}{\sum_{v_s \in e_j} \exp(\mathbf{r}_{js}^{l-1})} \quad (6)$$

where W_c and W_s are trainable matrices, while $\sigma(\cdot)$ is the non-linear activation function (e.g., ReLU).

In particular, the information of constructions is injected to hyperedges with trainable matrix W_g as:

$$\mathbf{g}_j^l = \mathbf{g}_j^l + W_g \mathbf{z}_j \quad (7)$$

Edge-level attention. As an inverse procedure, we fuse the information of the hyperedges back to each node via the edge-level attention. Formally:

$$\mathbf{h}_i^l = \sum_{e_j \in \mathcal{E}_i} \beta_{ij} W_e \mathbf{g}_j^l$$

$$\mathbf{t}_{ij}^{l-1} = \sigma((W_o \mathbf{h}_i^{l-1})^T W_r \mathbf{g}_j^l)$$

$$\beta_{ij} = \frac{\exp(\mathbf{t}_{ij}^{l-1})}{\sum_{e_j \in \mathcal{E}_i} \exp(\mathbf{t}_{ij}^{l-1})} \quad (8)$$

where W_e , W_o and W_r are trainable matrices. β_{ij} is the attention score of hyperedge e_j on node v_i .

After the enhanced representation \mathbf{H}^l is acquired with mutual attention mechanism in R-HGAT network, we apply the feed-forward network (FFN)

consisting of two fully-connected layers coupled with residual connections to generate final node representation \mathbf{H}^l , it can be formulated as:

$$\mathbf{H}^l = \text{LN}(\mathbf{H}^{l-1} + \sigma(W_2(W_1 \mathbf{H}^l + b_1) + b_2)) \quad (9)$$

where W_1 and W_2 refer to trainable weight matrices, while b_1 and b_2 are the biases. Besides, $\text{LN}(\cdot)$ denotes the layer normalization operation.

3.4 Model Training

To apply representations to downstream tasks, we utilize average-pooling to obtain the task-specific representation \mathbf{h}_z , which retains most of the information on node representations. For classification tasks (e.g., sentiment analysis and paraphrase tasks), \mathbf{h}_z is passed to a fully connected softmax layer, where the objective function to be minimized is the cross-entropy loss. As for regression task (i.e., similarity task), mean squared error loss is adopted as the objective function for optimization.

4 Experiments

4.1 Experiments Setup

Datasets. Experiments are conducted on a variety of NLU tasks for evaluation. We adopt ABSA datasets from SemEval and MAMS (Pontiki et al., 2014, 2015, 2016; Jiang et al., 2019) as well as GLUE (Wang et al., 2018) benchmark to demonstrate the effectiveness of our HyCxG. GLUE contains a broad range of datasets, including natural language inference (MNLI, QNLI, RTE), sentence-level sentiment analysis (SST-2), paraphrase and similarity (MRPC, QQP, STS-B), and linguistic ac-

Model	Rest14		Lap14		Rest15		Rest16		MAMS	
	Acc	F ₁	Acc	F ₁	Acc	F ₁	Acc	F ₁	Acc	F ₁
BERT-SPC	85.09	78.43	79.47	75.90	83.21	61.43	90.74	74.54	82.34	81.94
LCFS-ASC	86.71	80.31	80.52	77.13	82.47	66.39	89.77	75.23	82.78	82.25
R-GAT	86.60	81.35	78.21	74.07	84.32	68.47	91.56	75.85	84.13	83.78
KumaGCN	86.43	80.30	81.98	78.81	86.35	70.76	92.53	79.24	84.37	83.83
DGEDT	86.30	80.00	79.80	75.60	84.00	71.00	91.90	79.00	84.21	83.65
dotGCN	86.16	80.49	81.03	78.10	85.24	72.74	93.18	82.32	84.95	84.44
DualGCN	87.13	81.16	81.80	78.10	84.50	71.65	91.72	79.46	84.51	84.18
HyCxG	87.32	82.24	82.29	79.11	86.16	74.63	93.83	82.27	85.03	84.40

Table 2: Experimental results on ABSA datasets with BERT encoder. The best result on each dataset is in **bold**.

ceptability (CoLA). The detailed statistics of these datasets are provided in Appendix B.

Implementation. To solve Cond-MC problem, we set the weight scores to 1.0, 1.2 and 1.5 for s_{syn} , s_{sem} and s_{lex} (Equation 1), while w_1 , w_2 and w_3 are set to 1.0, 0.4 and 0.3 (Equation 2). Meanwhile, the COOL(\cdot) function is formulated as $T_0 \cdot a^k$, while a is the factor calculated as $-\ln(T_0/T_f)$.

HyCxG is optimized with AdamW (Loshchilov and Hutter, 2018) optimizer. The optimal hyperparameters are selected when the model achieves the highest performances on the development set via grid search. More detailed setups are described in Appendix E, and the pre-trained model weights are obtained from Hugging Face (Wolf et al., 2020).

Baselines. We compare HyCxG with BERT-based state-of-the-art baselines on ABSA datasets: (1) **BERT-SPC** denotes the fine-tuning BERT with aspect-specific representation. (2) **LCFS-ASC** (Phan and Ogunbona, 2020) employs the syntactic distance to alleviate the interference from unrelated words. (3) **R-GAT** (Wang et al., 2020) utilizes a relational graph attention network to encode the pruned dependency tree. (4) **KumaGCN** (Chen et al., 2020) synthesizes the dependency tree and latent graphs to enhance representation. (5) **DGEDT** (Tang et al., 2020) provides a dependency enhanced dual-transformer for classification. (6) **DualGCN** (Li et al., 2021) links the syntactic structure and semantic relevance to generate features. (7) **dotGCN** (Chen et al., 2022) builds the induced tree with syntactic distances to encode opinion information.

For other NLU tasks from the GLUE benchmark, we compare HyCxG with the base and large versions of the RoBERTa (Liu et al., 2019) model to demonstrate the performance improvement.

4.2 Experimental Results

Results on ABSA tasks. Experiments on ABSA tasks are conducted firstly, which are intuitive to analyze the function of constructions with aspects. Meanwhile, the results can be extended to multilingual settings as well as accessing to case studies. The model performances are shown in Table 2, from which several observations can be obtained.

First, compared to BERT-SPC model, most of syntax-based baselines achieve higher performances on five datasets, since they can alleviate interference introduced and build the relationship between aspects and their corresponding opinion words via synthesizing syntactic knowledge (e.g., dependency tree). Second, the baseline models (i.e., DualGCN and dotGCN) with more information, such as semantic and opinion representations, gain better performances than others.

Third, our HyCxG significantly outperforms all baselines with constructional information incorporated, which demonstrates the effectiveness of HyCxG. Furthermore, the perspective that construction grammar can contribute to sentence comprehension for NLU tasks can also be substantiated.

Results on GLUE tasks. As shown in Table 3, we conduct performance comparison among HyCxG with the base (B) and large (L) version of RoBERTa on GLUE development sets. And the results on test set are reported via online leaderboard.

The average results over tasks illustrate that our HyCxG_B outperforms RoBERTa_B, and HyCxG_L is also better than RoBERTa_L on both development and test sets. It indicates the validity of HyCxG and the benefits of constructional information.

As a branch of grammar, HyCxG has huge improvement on CoLA task, which shows its poten-

Model	CoLA Mcc	SST Acc	MNLI Acc	QNLI Acc	RTE Acc	QQP Acc	MRPC Acc	STS Pear	Avg.	
• Results on GLUE development set										
RoBERTa _B	63.6	94.8	87.6	87.5	92.8	78.7	91.9	90.2	91.2	86.5
RoBERTa _L	68.0	96.4	90.2	90.2	94.7	86.6	92.2	90.9	92.4	89.1
HyCxG _B	64.9	95.4	87.8	87.7	93.1	80.9	91.9	90.9	91.7	87.1
HyCxG _L	69.6	97.1	90.8	90.4	95.0	89.5	92.3	91.9	92.6	89.9
• Results on GLUE test set (from online leaderboard as of Dec., 2022)										
RoBERTa _B	60.3	95.7	87.5	87.2	93.0	79.7	89.5	88.1	89.5	85.6
RoBERTa _L	63.0	96.3	89.9	89.4	94.5	85.2	89.6	88.5	91.4	87.5
HyCxG _B	61.6	96.0	87.7	87.4	93.2	81.2	89.5	88.6	90.7	86.2
HyCxG _L	65.9	96.7	90.5	89.9	94.7	86.4	89.7	90.0	91.9	88.4

Table 3: Results for NLU tasks on GLUE development and test set. The best result on each task is in **bold**. **Mcc** refers to Matthews correlation coefficient, and **Pear** refers to Pearson correlation. **MNLI** task consists of the *matched* and *mismatched* datasets, while **SST** and **STS** denote the SST-2 and STS-B datasets, respectively.

Model	Rest14	Lap14	Rest15	Rest16
OML	85.80	80.25	84.32	91.56
OML+Pretrain	86.25	81.35	85.06	92.05
OLL+HyCxG	86.34	80.88	84.69	92.21
HyCxG	87.32	82.29	86.16	93.83

Table 4: Accuracy comparisons between HyCxG, OML and OLL+HyCxG on ABSA tasks with BERT encoder.

tial capability for evaluating linguistic acceptability. Meanwhile, HyCxG also significantly outperforms RoBERTa on RTE and MRPC tasks. Especially on the MRPC task, HyCxG_B even surpasses RoBERTa_L, which further validates the language representation can be enhanced by constructional information. Besides, HyCxG also boosts the performances on all other NLU tasks as well.

4.3 Comparative Analysis

Ablation study on model complexity. Though the efficiency of our HyCxG model is illustrated from the results on NLU tasks, we conduct ablation study to further investigate whether the performance enhancement is due to an increase in model complexity. We implement OML and OLL+HyCxG to address this concern. As increasing the depth of the PLMs can improve performance, OML adds an additional transformer layer on BERT, while OLL+HyCxG removes the last layer of BERT in HyCxG. Besides, we continue pre-training OML model (OML+Pretrain) on the massive corpus to achieve its optimal performance.

As shown in Table 4, OML achieves higher re-

Strategy	Rest14	Lap14	Rest15	Rest16
Cond-MC	87.32	82.29	86.16	93.83
w/o Cond	85.98	80.41	85.24	92.86
w/o Selection	86.43	81.03	84.32	92.05

Table 5: Results for different strategies of selecting constructions on four SemEval datasets.

sults than BERT-SPC, indicating that more complicated models tend to yield better performance. OML+Pretrain performs better compared to OML, which indicates the effect of pre-training. However, OLL+HyCxG surpasses OML with lack of two transformer layers, while it also overtakes OML+Pretrain on Rest 14 and 16 datasets. Meanwhile, HyCxG significantly outperforms OML and OML+Pretrain. As the details of computation complexity comparison between HyCxG and other baselines shown in Appendix D, the results illustrate that HyCxG achieves the highest performance at a relatively low computational complexity. These evidences all prove the efficacy of HyCxG and constructional information. The ablation study on GLUE benchmark is conducted in Appendix G.

Comparison of construction selection strategies.

In section 2, we formulate the Cond-MC problem for selecting typical constructions. Table 5 illustrates the performances for different strategies.

The strategy of w/o Cond denotes that constructions cannot overlap with each other, while w/o Selection employs all constructions. As redundancy and imbalance can occur in w/o Selection and w/o

Model	French	Spanish	Turkish	Dutch
BERT-SPC	83.15	90.42	91.72	87.56
R-GAT	83.84	90.78	92.41	88.32
KumaGCN	83.70	91.79	91.72	88.07
DGEDT	83.43	91.34	93.10	88.32
DualGCN	83.98	91.24	92.41	88.58
HyCxG	84.54	92.18	94.48	89.34

Table 6: Accuracy comparisons on SemEval multilingual datasets with BERT encoder.

Cond, w/o Cond performs better on Rest 15 and 16 datasets, while fails on Rest 14 and Lap 14 datasets compared to w/o Selection. In contrast, Cond-MC strategy consistently outperforms w/o Selection and w/o Cond on all datasets. The results illustrate the necessity of selecting typical constructions and the validity of Cond-MC strategy.

Multilingual results. Since construction grammar is applicable to multilingual analysis and can be learned via a unified framework for different languages (Dunn, 2022), we can further discuss the performance improvements of our HyCxG model in multilingual settings. The ABSA datasets in SemEval 2016 (Pontiki et al., 2016) with multiple languages are conducted for our experiments (statistics are shown in Table 8). Meanwhile, we also adapt baselines to multilingual environment with their official implementations. For models that involve dependency tree parsing (e.g., R-GAT, DualGCN), we maintain the same parsing tools that they employ (spaCy or Stanford Parser (Manning et al., 2014)). The results are shown in Table 6. We can observe that the inclusion of syntactic information can also improve model performances on multilingual settings compared to BERT-SPC. DualGCN has relatively high performance on different languages, since it incorporates the semantic features via SemGCN to the syntactic foundation with regularizers. Moreover, HyCxG outperforms these baselines across all four languages, indicating that constructional information can also enhance semantic understanding for other languages.

More empirical studies. We conduct experiments to investigate the potential capacity of HyCxG. First, experiments in Appendix F demonstrate the superior ability of HyCxG in perceiving statements and patterns. Second, we conduct ablation study in Appendix I to examine the benefits for each component in HyCxG. We also investigate the results on colloquial datasets in Appendix H.

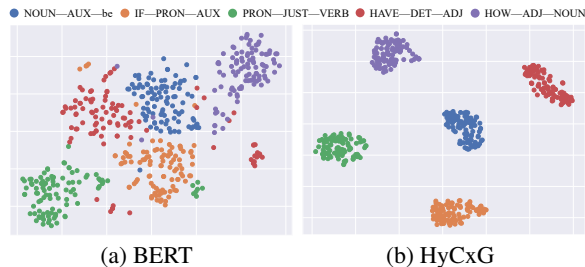


Figure 2: 2D t-SNE plot of construction representations.

As the constructions are derived from a formal language corpus, which has a different register (Dunn, 2023) than the colloquial dataset, some of the edge constructions are not extracted to enhance the representation, resulting in suboptimal performances. It illustrates the necessity of obtaining constructions on a diverse corpus. Besides, we present an approach in Appendix J to build the construction network with the representations of constructions acquired in HyCxG. It depicts the inheritance relations between constructions, which provides meaningful implications to usage-based CxG.

4.4 Case Study

The case study is utilized to illustrate the mechanism inherent in HyCxG for NLU tasks. We first visualize the learned representations via t-SNE (Van der Maaten and Hinton, 2008) in Figure 2, while the representations of these five randomly selected constructions are obtained via applying average-pooling in BERT-SPC and HyCxG. The results demonstrate that the representations in HyCxG form clusters with distinct boundaries, while the representations in BERT-SPC are diffuse among different clusters. It proves that HyCxG can model higher-order interactions and synthesize constructional information.

Meanwhile, we visualize the attention scores to illustrate the benefits of constructional information in Figure 3. As discussed in Section 1, BERT-SPC mainly focuses on the opinion word “friendly”, leading to misclassification. In contrast, HyCxG can produce correct prediction, since it captures the modality pattern (i.e., NOUN-AUX-be).

5 Related Work

Applications of CxG in NLP. CxG theories have been explored to NLP applications. Doubleday et al. (2017) exploit embodied CxG to parse the text of requests from users, while Nevens et al. (2019) employ fluid CxG for semantic parsing in

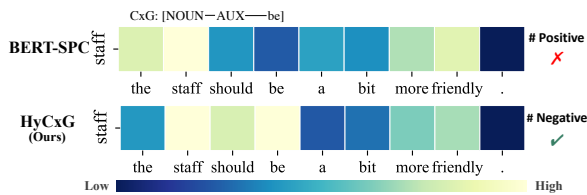


Figure 3: Visualization of attention scores from BERT-SPC and our HyCxG. Sentiment polarities after the marker # refer to the prediction label of each model.

visual question answering. Besides, a knowledge network is generated (Kiselev, 2020) with CxG on winograd schema challenge. These approaches all rely on hand-made definition with prior knowledge and are not compatible with PLMs, thus we employ the usage-based CxG in our work. However, we observe the problems of redundancy and imbalance in usage-based approach, which motivates us to formulate the conditional max coverage problem to acquire discriminative constructions.

Probing CxG in PLMs. Recent work has conducted empirical studies to probe whether PLMs acquire constructions. CxGBERT (Tayyar Madabushi et al., 2020) finds that constructional information is accessed by BERT but cannot be exploited for lack of proper methods. CxLM (Tseng et al., 2022) shows that while PLMs are aware of the construction, they are confused at the variable slots. Weissweiler et al. (2022) discover that PLMs can recognize the comparative correlative constructions but fail to utilize their meaning. Furthermore, Weissweiler et al. (2023) argue that fine-tuning on downstream tasks necessitates explicit access to constructional information. Our work is the first attempt to exploit the constructional information for representation enhancement on NLU tasks.

Hypergraph neural networks. Recent research has extended conventional graph to hypergraph (Feng et al., 2019; Ding et al., 2020a; Cai et al., 2022), which can model high-order data correlation. However, when the networks ignore the representation of hyperedges, the capability of capturing higher-order relationships between nodes can be inhibited (Fan et al., 2021). It motivates us to propose R-HGAT to encode constructional information.

Aspect-based sentiment analysis. There has been much work on syntax-based methods (Wang et al., 2020; Tang et al., 2020; Phan and Ogunbona, 2020; Li et al., 2021; Tang et al., 2022; Xu et al., 2023), which establish syntactic connections between each aspect and their corresponding opinion

words via dependency and constituency parsing.

6 Conclusion

In this work, we introduce usage-based construction grammar (CxG) to enrich language representation with constructional information. Then a hypergraph framework of CxG named HyCxG is proposed to integrate constructional information for language representation enhancement. In HyCxG, we extract constructions through a slot-constraint method, and formulate the conditional max coverage problem for selecting the discriminative constructions. Then we propose a relational hypergraph attention network to encode high-order word interactions within constructions. Extensive experiments illustrate the validity of our HyCxG on natural language understanding tasks.

Limitations

In this study, the limitations can be summarized into two major aspects:

(1) The usage-based approach (Dunn, 2017, 2019) being employed in our work has the ability to extract most of constructions, while a small portion of non-contiguous constructions (e.g., comparative correlative constructions) are neglected. These non-contiguous constructions are probably fragmented into multiple independent constructions. We will investigate the approaches to capture these non-contiguous constructions via incorporating more syntactic knowledge in future work for language representation enhancement.

(2) As discussed in Appendix H, the performances are not significantly improved on the tasks that contain a large amount of colloquial expressions. Since our constructions are mainly learned from the formal corpus, which has less colloquial expressions. It causes fewer constructions to be accessed in these tasks, which encourages us to learn constructions in more diverse corpus to enhance the language representation for natural language understanding tasks in the future.

Acknowledgements

We thank all anonymous reviewers for their insightful comments and suggestions. And we also appreciate High-Flyer for providing us with computational cards for pre-training. This research is supported by the Science and Technology Project of Zhejiang Province (2022C01044) and the National Natural Science Foundation of China (51775496).

References

- Giulia ML Bencini and Adele E Goldberg. 2000. [The contribution of argument structure constructions to sentence meaning](#). *Journal of Memory and Language*, 43(4):640–651.
- Ronny Boogaart, Timothy Colleman, and Gijsbert Rutten. 2014. *Extending the scope of construction grammar*. De Gruyter Mouton.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. [Language models are few-shot learners](#). *Advances in neural information processing systems*, 33:1877–1901.
- Derun Cai, Moxian Song, Chenxi Sun, Baofeng Zhang, Shenda Hong, and Hongyan Li. 2022. [Hypergraph structure learning for hypergraph neural networks](#). In *IJCAI*, volume 7, pages 1923–1929.
- Chenhua Chen, Zhiyang Teng, Zhongqing Wang, and Yue Zhang. 2022. [Discrete opinion tree induction for aspect-based sentiment analysis](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2064. Association for Computational Linguistics.
- Chenhua Chen, Zhiyang Teng, and Yue Zhang. 2020. [Inducing target-specific latent structures for aspect sentiment classification](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5596–5607. Association for Computational Linguistics.
- 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*, pages 4171–4186. Association for Computational Linguistics.
- Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020a. [Be more with less: Hypergraph attention networks for inductive text classification](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936. Association for Computational Linguistics.
- Xiao Ding, Dingkui Hao, Yuewei Zhang, Kuo Liao, Zhongyang Li, Bing Qin, and Ting Liu. 2020b. [HIT-SCIR at SemEval-2020 task 5: Training pre-trained language model with pseudo-labeling data for counterfactuals detection](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 354–360. International Committee for Computational Linguistics.
- Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. [Adaptive recursive neural network for target-dependent Twitter sentiment classification](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54. Association for Computational Linguistics.
- Steve Doubleday, Sean Trott, and Jerome Feldman. 2017. [Processing natural language about ongoing actions in eeg](#). In *2017 aaii spring symposium series*.
- Jonathan Dunn. 2017. [Computational learning of construction grammars](#). *Language and cognition*, 9(2):254–292.
- Jonathan Dunn. 2019. [Frequency vs. association for constraint selection in usage-based construction grammar](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 117–128, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jonathan Dunn. 2022. [Exposure and emergence in usage-based grammar: computational experiments in 35 languages](#). *Cognitive Linguistics*.
- Jonathan Dunn. 2023. [Exploring the construction: Linguistic analysis of a computational CxG](#). In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 1–11, Washington, D.C. Association for Computational Linguistics.
- Jonathan Dunn and Sidney Wong. 2022. [Stability of syntactic dialect classification over space and time](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 26–36. International Committee on Computational Linguistics.
- Vyvyan Evans. 2007. *Glossary of cognitive linguistics*. Edinburgh University Press.
- Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. 2021. [Heterogeneous hypergraph variational autoencoder for link prediction](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. [Hypergraph neural networks](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565.
- Adele Goldberg. 1995. *Constructions: a construction grammar approach to argument structure*. Chicago: The University of Chicago.
- Adele Goldberg. 2006. *Constructions at work: Constructionist approaches in context*. Oxford University Press on Demand.
- Adele E Goldberg. 2003. [Constructions: A new theoretical approach to language](#). *Trends in cognitive sciences*, 7(5):219–224.

- Adele E Goldberg, Mirjam Fried, and Jan-Ola Östman. 2005. *Construction grammar (s): Cognitive and cross-language dimension*. John Benjamins Amsterdam.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *LREC 2018*. European Language Resources Association.
- Stefan Th Gries and Stefanie Wulff. 2005. Do foreign language learners also have constructions? *Annual Review of Cognitive Linguistics*, 3(1):182–200.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [Deberta: Decoding-enhanced bert with disentangled attention](#). In *International Conference on Learning Representations*.
- Martin Hilpert. 2014. *Construction grammar and its application to English*. Edinburgh University Press.
- Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford handbook of construction grammar*. Oxford University Press.
- Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. [A challenge dataset and effective models for aspect-based sentiment analysis](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6280–6285.
- Vsevolod Kapatsinski. 2014. [What is grammar like? a usage-based constructionist perspective](#). In *Linguistic Issues in Language Technology, Volume 11, 2014 - Theoretical and Computational Morphology: New Trends and Synergies*. CSLI Publications.
- Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. [Optimization by simulated annealing](#). *science*, 220(4598):671–680.
- Denis Kiselev. 2020. An ai using construction grammar: Automatic acquisition of knowledge about words. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 289–296.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.
- Ronald W Langacker. 1987. *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford university press.
- Ronald W Langacker. 2005. [Constructing a language: A usage-based theory of language acquisition](#). *Language*, 81(3):748–750.
- Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.
- Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. [Dual graph convolutional networks for aspect-based sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6319–6329. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2018. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations (ICLR)*.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.
- Jens Nevens, Paul Van Eecke, and Katrien Beuls. 2019. [Computational construction grammar for visual question answering](#). *Linguistics Vanguard*, 5(1).
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. [A universal part-of-speech tagset](#). In *LREC’12*, pages 2089–2096. European Language Resources Association.
- Minh Hieu Phan and Philip O. Ogunbona. 2020. [Modelling context and syntactical features for aspect-based sentiment analysis](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3220. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. [SemEval-2016 task 5: Aspect based sentiment analysis](#). In *SemEval-2016*, pages 19–30.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. [SemEval-2015 task 12: Aspect based sentiment analysis](#). In *SemEval 2015*, pages 486–495.

- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [SemEval-2014 task 4: Aspect based sentiment analysis](#). In *SemEval 2014*, pages 27–35.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21(140):1–67.
- Giulia Rambelli, Emmanuele Chersoni, Philippe Blache, Chu-Ren Huang, and Alessandro Lenci. 2019. [Distributional semantics meets construction grammar: towards a unified usage-based model of grammar and meaning](#). In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 110–120. Association for Computational Linguistics.
- Luc Steels and Joachim de Beule. 2006. [A \(very\) brief introduction to fluid construction grammar](#). In *Proceedings of the Third Workshop on Scalable Natural Language Understanding*, pages 73–80, New York City, New York. Association for Computational Linguistics.
- Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. [Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588. Association for Computational Linguistics.
- Siyu Tang, Heyan Chai, Ziyi Yao, Ye Ding, Cuiyun Gao, Binxing Fang, and Qing Liao. 2022. [Affective knowledge enhanced multiple-graph fusion networks for aspect-based sentiment analysis](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5352–5362. Association for Computational Linguistics.
- Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. [CxGBERT: BERT meets construction grammar](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032. International Committee on Computational Linguistics.
- Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. [CxLM: A construction and context-aware language model](#). In *LREC’22*, pages 6361–6369, Marseille, France. European Language Resources Association.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-sne](#). *Journal of machine learning research*, 9(11).
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. [Graph attention networks](#). *arXiv preprint arXiv:1710.10903*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.
- Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. [Relational graph attention network for aspect-based sentiment analysis](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238. Association for Computational Linguistics.
- Leonie Weissweiler, Taiqi He, Naoki Otani, David R. Mortensen, Lori Levin, and Hinrich Schütze. 2023. [Construction grammar provides unique insight into neural language models](#). In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 85–95, Washington, D.C. Association for Computational Linguistics.
- Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. [The better your syntax, the better your semantics? probing pretrained language models for the english comparative correlative](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.
- Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. [Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback](#). *GermEval*, pages 1–12.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.
- Lvxiaowei Xu, Xiaoxuan Pang, Jianwang Wu, Ming Cai, and Jiawei Peng. 2023. [Learn from structural scope: Improving aspect-level sentiment analysis with hybrid graph convolutional networks](#). *Neurocomputing*, 518:373–383.
- Xiaoyu Yang, Stephen Obadinma, Huasha Zhao, Qiong Zhang, Stan Matwin, and Xiaodan Zhu. 2020. [SemEval-2020 task 5: Counterfactual recognition](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 322–335. International Committee for Computational Linguistics.

A Background

A.1 Construction Grammar

Construction Grammar (CxG) theory is a branch of cognitive linguistics. It assumes that grammar is a meaningful continuum of lexicon, morphology and syntax rather than solely relying on a system composed of stable but arbitrary rules for generating well-formed sequences. Therefore, constructions (i.e., symbolic elements that connect a certain morphosyntactic form to a meaning) are thought to be the primary objects of grammatical description (Langacker, 1987; Goldberg, 1995, 2006). Constructions can be defined as linguistic patterns that store different form and meaning pairs. They cannot be strictly predictable by their components. (Goldberg, 1995; Langacker, 2005). Besides, patterns that occur with sufficient frequency can also be considered as constructions (Goldberg, 2006).

In terms of the form for CxG, their syntactic structure varies in level of abstractness, including partially or fully filled components (e.g., idioms), and general linguistic patterns (Goldberg, 2003, 2006; Dunn and Wong, 2022). In another perspective, each construction has immutable components and unfilled slots that the element in the slots can be altered with certain words to boost productivity.

Meanwhile, constructions can also be regarded as linguistic knowledge. This knowledge is based on cognitive results and the product of empirical generalization and abstraction. In the evolution of human language experience, the knowledge is formed, acquired and applied by our human (Goldberg, 2003; Hilpert, 2014). Analogy to Knowledge Graph (KG), the language systems are considered as a network of constructions (Goldberg, 1995). This network consists of two fundamental components: (1) nodes in the graph (i.e., specific constructions). (2) the edges between nodes. These edges refer to diverse relationships between constructions (e.g., polysemy link, metaphorical extension and instance link). It is worth mentioning that there may exist multiple connections between constructions in the network.

A.2 Usage-based Construction Grammar

The CxG paradigm has already developed a variety of implementations, including formal approaches, i.e., fluid construction grammar (FCG), embodied construction grammar (ECG) and sign-based construction grammar (SBCG) as well as usage-based approaches. However, the FCG, ECG and

Dataset	$ L $	#Train	#Test	#Task
• Single-sentence task				
CoLA	2	8,551	1,063	Acceptability
SST-2	2	67,349	1,821	Sentiment
• Sentence-pair task				
QQP	2	363,846	390,965	Paraphrase
MRPC	2	3,668	1,725	Paraphrase
MNLI	3	392,702	19,643	NLI
QNLI	2	104,743	5,463	NLI
RTE	2	2,490	3,000	NLI
STS-B	*	5,749	1,379	Similarity

Table 7: Statistics for the GLUE benchmark. $|L|$ indicates the number of classes for classification tasks, while * refers to regression task. The labels of STS-B are continuous values from 0 to 5.

SBCG rely heavily on hand-made definition with prior knowledge (Steels and de Beule, 2006; Goldberg et al., 2005; Rambelli et al., 2019). Though these methods can provide high-quality representations, they cannot automatically mine the constructions from the data with the emergence of slot-constraints (Dunn, 2017, 2019).

We follow the efforts of Dunn (2019) that propose a computationally slot-constraints approach for CxG extraction via the data-driven pipeline. First, three types of slots (i.e., lexical, syntactic and joint semantic-syntactic) are defined according to the different levels of abstraction. Note that the syntactic slots refer to the POS tags, while the semantic-syntactic slots are derived from word embeddings that are clustered in discrete semantic domains based on K-Means algorithm (Dunn and Wong, 2022). Secondly, they generate the potential construction templates from the corpus. Then the statistics of frequency and association strength are calculated for pruning templates. Finally, the tabu search is applied to determine the optimal set of constructions with Minimum Description Length (MDL) metric (Dunn, 2019). Based on this pipeline, they develop the c2xg² toolkit.

B Data Statistics

We conduct experiments on a variety of natural language understanding tasks in main experiments, including eight tasks from the GLUE benchmark (Wang et al., 2018) and five aspect-level sentiment analysis tasks (Pontiki et al., 2014, 2015, 2016; Jiang et al., 2019). Table 7 presents the

²<https://pypi.org/project/c2xg/>

Dataset	Division	#Pos.	#Neu.	#Neg.
• Main Experiments				
MAMS	Train	3,380	5,042	2,764
	Develop	403	604	325
	Test	400	607	329
Rest 14	Train	2,164	637	807
	Test	728	196	196
Lap 14	Train	994	464	870
	Test	341	169	128
Rest 15	Train	912	36	256
	Test	326	34	182
Rest 16	Train	1,240	69	439
	Test	469	30	117
• Multilingual Experiments				
French	Train	901	116	753
	Test	364	69	285
Spanish	Train	1,368	89	479
	Test	521	34	176
Turkish	Train	770	111	504
	Test	93	6	46
Dutch	Train	758	118	407
	Test	245	24	125
• Colloquial Experiments				
Twitter	Train	1,561	3,127	1,560
	Test	173	346	173
GermEval	Train	1,592	812	4,554
	Develop	207	83	777
	Test	127	175	1,112

Table 8: Statistics for ABSA datasets that be employed in the main, multilingual and colloquial experiments.

data statistics for GLUE benchmark tasks. We divide the GLUE benchmark tasks into two categories: single-sentence task and sentence-pair task. In single-sentence task, the language acceptability task (CoLA) and the sentence-level sentiment task (SST-2) are included. As for sentence-pair task, it contains similarity task (STS-B), paraphrasing tasks (MRPC, QQP) and several natural language inference (NLI) tasks (MNLI, QNLI and RTE). All tasks except STS-B are classification tasks, while STS-B is a regression task.

The statistics for aspect-based sentiment analysis datasets are shown in Table 8. They are all annotated with aspects and corresponding polarities (i.e., positive, neutral or negative). In multilingual settings, we adopt four different language datasets from SemEval 2016 (Pontiki et al., 2016). Meanwhile, Twitter (Dong et al., 2014) and GermEval (Wojatzki et al., 2017) datasets, which collect tweets and messages from social media, are employed in the colloquial expression experiments.

C Learning and Alignment of CxG

In order to utilize constructional information, the inventory of construction grammar needs to be learned at first. As we present a brief description of construction learning procedure in Section A.2, prior work has shown that construction grammars can converge on stable representations with sufficient training data (Dunn, 2022). Therefore, the construction grammar in English is learned from sampling of multi-source corpus (WIKIPEDIA, BOOKCORPUS and CC-NEWS) (Devlin et al., 2019; Liu et al., 2019) which contains about 1,200 million words. As for other languages utilized in multilingual experiments, constructions are learned from sampling the multilingual portion of the C4 corpus (Raffel et al., 2020). Since the constructions on a variety of languages are learned in c2xg toolkit, we validate that our HyCxG can achieve comparable performances on NLU tasks with the construction grammar list in c2xg.

After obtaining the construction grammar list, we are able to derive all possible constructions from a given sentence via c2xg toolkit. However, the toolkit can only detect which construction patterns are present in the sentences and cannot provide specific position indexes of them. Besides, there are discrepancies between c2xg and pre-trained language models in tokenization procedure. The c2xg toolkit utilizes the basic white-space tokenizer, while pre-trained language models adopt more complex algorithms (e.g., WordPiece (Devlin et al., 2019), SentencePiece (Kudo and Richardson, 2018)). Therefore, we have to align the position indexes of the constructions under different tokenization algorithms with specific mapping function.

To tackle these imperatives, we develop a wrapper for c2xg. It can be adapted to different tokenization methods with the output of the start and end position index for construction spans. We hope that it can facilitate future research on construction grammar in natural language processing tasks.

D Analysis of Computational Complexity

In Table 4, we demonstrate that the effectiveness of our model does not result from the increased complexity of the model. As shown in Table 9, we further analyze the computational complexity of the models via DEEPSPEED to illustrate more intuitive results. The number of model parameters and the multiply-accumulate operations (MACs) are utilized to compare complexity on aspect-based

Model	Params	w/o PLM	MACs
BERT-SPC	109.49M	0.01M	13.16G
R-GAT	110.45M	0.97M	13.20G
LCFS-ASC	228.42M	9.46M	26.79G
KumaGCN	112.49M	3.01M	13.87G
DGEDT	112.50M	3.02M	14.08G
DualGCN	111.85M	2.37M	13.74G
OML	116.57M	7.09M	14.25G
HyCxG	112.24M	2.76M	13.48G

Table 9: Computational complexity analysis of models. **Params**, **w/o PLM** and **MACs** represent the number of parameters, model parameters without PLM and multiply-accumulate operations, respectively.

sentiment analysis models. To ensure a fair comparison, we unify the public hyper-parameters of all baselines while preserving other unique hyper-parameters consistent with their official implementations. Besides, the additional embedding parameters for constructions in HyCxG are not counted. Then three observations can be derived.

First, LCFS-ASC is significantly more complex than other models due to its dual BERT architecture, while other models all use a single PLM. Second, adding an additional transformer layer to BERT (OML) indeed improves performances but is far more complex than our HyCxG. However, our model outperforms OML, which proves the improvement of our HyCxG is not just due to the increased complexity of the model structure. Third, our model has lower computational cost and memory consumption than most of baselines. Meanwhile, our HyCxG achieves even better performances without dependency information injection, which empirically validates the effectiveness of constructional information.

E Hyper-parameters and Settings

For fine-tuning our HyCxG on GLUE and aspect-based sentiment analysis datasets, we search for the optimal task-specific hyper-parameters with the range of values in Table 10 (We implement our model in PYTORCH and use GeForce RTX 3090 devices for experiments):

In order to pre-train OML models in Table 4 and 12, we follow the same pre-training procedure of BERT and RoBERTa. OML models are implemented in PYTORCH and pre-trained on $8 \times$ Tesla A100 devices. We employ the officially released PLM checkpoints from Hugging Face³ to initial-

³<https://huggingface.co/models>

Configuration	Value
Epoch	4 / 10 / 50
Batch size	16 / 24 / 32
Sequence length	150 / 250 / 300
Learning rate	1e-5 / 2e-5 / 3e-5
Dropout (PLM)	[0.0, 0.5]
Dropout (R-HGAT)	[0.0, 0.5]
Layers	1
Weight decay	1e-1 / 1e-2
Warmup ratio	0.03 / 0.06

Table 10: Detailed hyper-parameter configurations.

Models	Ensemble	Precision	Recall	F ₁
HIT	✗	88.27	90.76	89.49
HIT+PL	✓	91.90	90.00	90.90
HyCxG	✗	94.09	93.83	93.96
HyCxG+PL	✗	95.21	94.24	94.72

Table 11: Results on counterfactual detection dataset. **PL** is the abbreviation for pseudo-labeling.

ize basic parameters (12 layers) in OML, while the parameters in additional transformer layer are initialized to the average of the preceding 12 layers. Besides, we set the batch size to 64 per GPU device, and the training steps to 400,000 during pre-training procedure with AdamW optimizer (learning rate is 5e-5 and warmup ratio is 0.1) and the corpus of WIKIPEDIA and BOOKCORPUS.

F Pattern Recognition Capability of CxG

In this work, we demonstrate that the construction grammar is capable of improving natural language understanding via infusing constructional information. However, can CxG only serve to enhance language representation? We explore more application scenarios for construction grammar based on this concern and a viable scene is pattern recognition. The identification of this scenario can be viewed as an investigation into whether the text contains particular specific structures (e.g., negation and counterfactual expression). We take counterfactual detection task as an example to verify the feasibility of the scenario.

The counterfactual dataset in SemEval 2020 (Yang et al., 2020) are adopted in this experiment with train (13,000 instances) and test (7,000 instances) sets. We employ Precision, Recall and F₁ score as the metrics to evaluate the performance. To better evaluate our method, we compare the performances with the best system in SemEval.

Model	CoLA	SST-2	RTE	MRPC
OML	64.24	95.07	79.06	90.44
OML+Pretrain	64.84	95.18	80.51	91.18
OLL+HyCxG	64.42	95.30	79.78	90.69
HyCxG	64.90	95.41	80.87	90.93

Table 12: Ablation study on four GLUE development datasets with RoBERTa-base encoder.

HIT (Ding et al., 2020b) is the state-of-the-art method that employs the ensemble model (combining the large version of BERT, RoBERTa and XLNet via weighted average for their probability predictions) with pseudo-labeling (Lee et al., 2013) strategy and 10-fold cross-validation. Table 11 shows the results and the above system is marked as HIT+PL. HIT is the single model of RoBERTa-large without pseudo-labeling. Our HyCxG model is also trained with 10-fold cross-validation based on RoBERTa-large, while HyCxG+PL incorporates pseudo-labeling strategy to HyCxG.

Surprisingly, we observe that our HyCxG significantly outperforms HIT system by 4.47 F_1 points. Even for HIT+PL (applying ensemble and pseudo-labeling), there is still a massive gap with HyCxG. Furthermore, our HyCxG+PL also has a 3.82 F_1 points boost compared to HIT+PL. It endorses the ability of our model in pattern recognition.

As discussed by Yang et al. (2020), the main challenge in counterfactual detection task is that existing models focus excessively on token level features while neglecting to understand statements. Most counterfactual cases are expressed with subjunctive mood to convey wishes, suggestions and demands. There is a practical example “*if I were asked to, I would be happy to talk to anyone.*”, which is misclassified by HIT model, while HyCxG can predict correctly. It is a counterfactual expression that regards the part guided by the conjunction “*if*” as the antecedent. Based on such hypothesis, the possible consequent is stated. In this sentence, our system can extract two critical construction “*if-PRON-were-VERB*” \rightarrow “*if I were asked*” and “*AUX-be-ADJ*” \rightarrow “*would be happy*” for capturing the counterfactual statements. Thus, HyCxG is capable of recognizing specific patterns via encoding constructional information.

Similarly, negation detection is also a recognition task relying on a series of special patterns. There are more application scenarios of CxG, which we leave those discussions to future work.

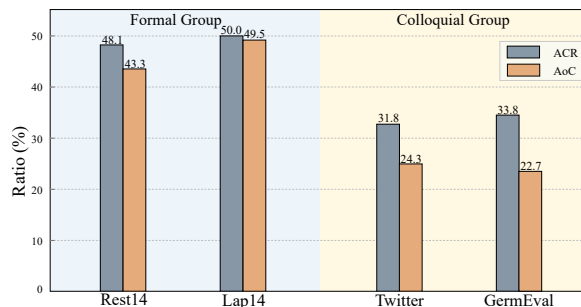


Figure 4: Comparison of construction sparsity based on AoC and ACR metrics.

G Ablation Study on GLUE Datasets

Similar to the ablation study on ABSA tasks (Table 4), we conduct experiments on GLUE benchmark with four datasets (CoLA, SST-2, RTE and MRPC). As discussed previously, the PLM (here is RoBERTa-base) with an additional layer on top is called OML. OML+Pretrain performs an extra pre-training procedure on OML (details of pre-training are shown in Section E). OLL+HyCxG is the model that removes the last transformer layer from PLM in our HyCxG model. For the results in Table 12, OML+Pretrain always outperforms OML on four tasks, which illustrates the utility of pre-training procedure. Besides, our OLL+HyCxG achieve better performances compared to OML, even though it has two fewer transformer layers. It is worth mentioning that the result of OLL+HyCxG is even higher than OML+Pretrain on SST-2 dataset.

Moreover, HyCxG performs better than OML and OML+Pretrain, while it has a smaller number of parameters with lower complexity. These results indicate the benefits of constructional information in natural language understanding tasks and high cost performance of our HyCxG.

H Colloquial Expression Results

As we discuss in Limitations, the performance improvement of HyCxG is not significant for datasets with informalized expressions (i.e., colloquial expressions). In this work, constructions are learned from the standard corpus (e.g., WIKIPEDIA and BOOKCORPUS) with formal language. This causes some of edge constructions not being extracted for language representation enhancement. We conduct experiments on Twitter (Dong et al., 2014) and GermEval (Wojatzki et al., 2017) datasets to investigate the issue for English and German. These datasets are collected from social media which consist of

Model	Twitter		GermEval	
	Acc	F ₁	Acc	F ₁
BERT-SPC	75.14	73.59	84.65	66.76
R-GAT	76.15	74.88	85.50	68.81
KumaGCN	77.89	77.03	85.08	69.26
DGEDT	77.90	75.40	86.71	72.81
DualGCN	77.40	76.02	86.96	73.44
HyCxG	77.17	75.70	86.85	71.71

Table 13: Experimental performances of datasets with colloquial expressions.

numerous colloquial sentences (statistics are shown in Table 8). As shown in Table 13, the results on Twitter are retrieved from the papers of baselines. Then we adapt all the models with multilingual settings (Section 4.3) to the GermEval dataset. From the results, we observe the performance of our HyCxG on Twitter is only in the middle compared to baselines. As for GermEval, HyCxG exceeds most of models but also falls below DualGCN.

To further analyze the evidence for supporting our conclusion, two metrics are proposed to evaluate the sparsity of constructions: average ratio of construction (AoC) and average coverage ratio (ACR). We define AoC as the average proportion of extracted constructions in each sentence, while ACR means the average ratio for the total length of the constructions to sequence length in each sentence under maximum coverage (Section 2.2). Metrics can also be formulated as follows:

$$\text{AoC} = \frac{1}{N} \sum_{i=1}^N \frac{\text{CxG}(s_i)}{\mathcal{L}(s_i)}$$

$$\text{ACR} = \frac{1}{N} \sum_{i=1}^N \frac{\sum_{j=1}^M \mathcal{L}(c_j)}{\mathcal{L}(s_i)}, c_j \in \text{MC}(s_i) \quad (10)$$

where N is the number of sentences in each dataset while s_i is the i -th sentence. $\mathcal{L}(\cdot)$ is the function, that computes the length of the input and $\text{CxG}(\cdot)$ is employed to calculate the total number of constructions in a sentence. Besides, $\text{MC}(\cdot)$ refers to the set of constructions in the sentence that satisfy the maximum coverage (w/o condition), while M is the size of the set and c_j is the j -th construction.

We compare these metrics between formal group (Rest14, Lap14) and colloquial group (Twitter, GermEval) and the results are shown in Figure 4. There is a huge gap, where colloquial group is significantly lower than formal group on AoC and ACR. It demonstrates that constructions are much sparser

Model	Rest14	Lap14	Rest15	Rest16
HyCxG	87.32	82.29	86.16	93.83
w/o HGATT	86.34	81.19	84.69	92.37
w/o FFN	86.70	81.50	85.06	93.02

Table 14: Experimental results of ablation study on components of HyCxG.

in Twitter and GermEval. In such circumstances, constructional information learned by our model has been inadequate so far to enhance the representation. For this limitation in our work, we believe that it can be improved via adding more colloquial corpus to the construction learning (Section C), which requires further investigations in the future.

I Ablation Study on Model Components

To further investigate the influence of different components in our HyCxG, we conduct extensive ablation studies. As shown in Table 14, w/o HGATT and FFN indicate that we temporarily remove each of these components from our model. The experimental results illustrate the importance of the HGATT network, since its removal can cause significant performance degradation. Meanwhile, it also demonstrates the validity of injecting constructional information for language representation enhancement. Overall, our HyCxG with all components can achieve the highest performance.

J Network of Constructions

Construction grammar is not an unordered set in a language, but rather a network linked by inheritance relations. The whole network of constructions in a particular language is called the *Constructicon* (Evans, 2007). As briefly introduced in Section A.1, *Constructicon* consists of nodes (constructions) and edges (inheritance relations) (Goldberg, 2003). There are four types of relations between constructions: polysemy link (I_P), subpart link (I_S), instance link (I_I) and metaphorical extension (I_M). Previous efforts are taken by linguists using case studies to manually build the networks (Boogaart et al., 2014). To the best of our knowledge, there is no attempt to construct *Constructicon* from the perspective of distributional representation. In this work, we can acquire representation of each construction via the embedding matrix in HyCxG. Therefore, our HyCxG can contribute to linguistics in turn to provide interpretability for the network of constructions.

Algorithm 2: Compute MD between constructions

Input: The set of constructions \mathcal{G} .
Output: MD matrix for \mathcal{G} .

```

1  $\mathcal{G} = [g_1, g_2, \dots, g_n]$ 
2 Initialize MD  $\in \mathbb{R}^{n \times n}$  matrix with 0
3 for  $i = 0$  to  $n - 1$  do
4   for  $j = 0$  to  $n - 1$  do
5     if  $i = j$  then
6       MD[ $i, j$ ]  $\leftarrow$  0
7     else
8       MD[ $i, j$ ]  $\leftarrow$  Compute_md( $g_i, g_j$ )
9     end
10  end
11 end
12 return MD
13
14 Function Compute_md( $A, B$ ):
15    $A = [c_1, c_2, \dots, c_p], B = [e_1, e_2, \dots, e_q]$ 
16   Initialize dp  $\in \mathbb{R}^{(p+1) \times (q+1)}$  matrix with 0
17   for  $i = 1$  to  $p + 1$  do
18     for  $j = 1$  to  $q + 1$  do
19       if Is_Match( $A[i-1], B[j-1]$ ) then
20         d  $\leftarrow$  0
21       else
22         d  $\leftarrow$  1
23       end
24       dp  $\leftarrow$  min(dp[ $i-1$ ][ $j$ ]+1, dp[ $i$ ][ $j-1$ ]+1, dp[ $i-1$ ][ $j-1$ ]+d)
25     end
26   end
27   dist  $\leftarrow$  dp[ $m$ ][ $n$ ]
28   return dist
29 end function

```

We discuss three types of inheritance relationships (I_P , I_S and I_I) in this work, while metaphorical extension is more abstract and complex. I_P characterizes the semantic relationship between a specific meaning of a construction and its extended meaning. I_S is referred to the connection that a construction is an independent subpart of another construction. Besides, when a concrete construction is a special instance of another construction, the relationship is defined as I_I . For the purpose of modeling inheritance relationships, we measure the distances between different constructions from semantic and morphological perspectives. Then we weight the sum of the two distances and the top- k closest constructions are selected for each construction to determine the relationships. Eventually, the network of constructions is generated.

Since the representation of each construction is obtained, semantic distance (SD) can be com-

Relation	Example
Polysemy	think-PRON-AUX
	believe-PRON-AUX
	know-PRON-AUX
Subpart	SCONJ-PRON-AUX-VERB-to
	SCONJ-PRON-AUX-VERB
	SCONJ-PRON-AUX
Instance	SCONJ-PRON-AUX
	if-PRON-AUX
	if-PRON-can

Table 15: Examples for three types of inheritance relationships in *construction*.

puted via cosine similarity. Meanwhile, we propose multi-level edit distance (MED) algorithm to evaluate the morphological distance (MD) between two constructions with different abstract level. In this setting, semantic distance can provide evidence for polysemy link, while morphological distance constrains variation in structures for subpart link and instance link. Thus, these distances are complementary on modeling inheritance relations. Under definition, semantic distance can be formulated as:

$$SD(g_i, g_j) = 1 - \frac{\vec{v}_{g_i} \cdot \vec{v}_{g_j}}{\|\vec{v}_{g_i}\| \|\vec{v}_{g_j}\|} \quad (11)$$

where \vec{v}_{g_i} and \vec{v}_{g_j} are the embedding vector of construction g_i and g_j , which can be looked up from construction embedding matrix E_c , respectively.

MED is proposed to compute the morphological distances. As slots in constructions are defined for different levels (Lexical, Syntactic and Semantic) of abstraction, we need to match slots across the levels. We define a function Is_Match(\cdot, \cdot) to indicate whether two slots match or not. The matching criteria can be summarized as follows:

- (Lexical \leftrightarrow Lexical) Match when the slots are identical with the same POS tags or belong to common semantic cluster.
- (Syntactic \leftrightarrow Syntactic, Semantic \leftrightarrow Semantic) Match if and only if the slots are identical.
- (Lexical \leftrightarrow Syntactic) Match when the slot of lexical has the same POS tags as the other.
- (Lexical \leftrightarrow Semantic) Match when the semantic cluster of the lexical slot is identical to the slot in semantic level.

Based on such matching strategy, MED algorithm is illustrated in Algorithm 2. After obtaining semantic and morphological distances, we synthesize them to compute the construction distance.

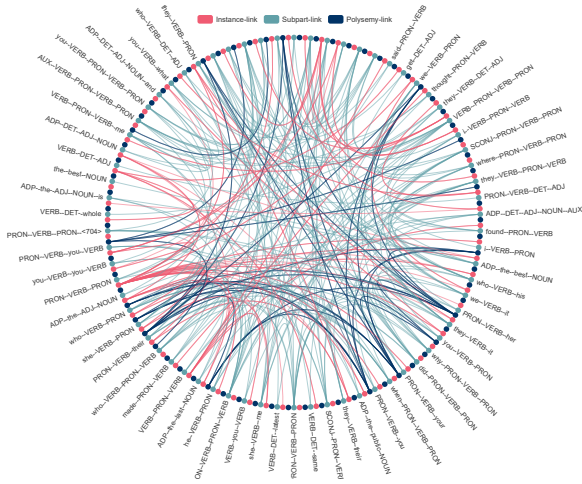


Figure 5: An example of a sub-network with three inheritance relationships with our visualization tool.

Then the top-k closest constructions for each construction can be selected as the potential inheritance set. It can be derived as:

$$\{\mathcal{G}_g \subset \mathcal{G} \mid \arg \min_{|\mathcal{G}_g|=k} (\gamma SD_g + MD_g)\} \quad (12)$$

where \mathcal{G}_g is the potential inheritance set for construction g , while \mathcal{G} is the set of all constructions. SD_g and MD_g refer to semantic and morphological distances between g and the other constructions in \mathcal{G} , respectively. γ is the coupling co-efficiency that regulates the two distances. Then we can determine the inheritance relationship within the set based on the definition of I_P , I_S and I_I .

In order to analyze *Constructicon*, the embedding matrix E_c of constructions is extracted after training on CoLA task as an example. We set k to 15 and γ to 10 in Equation 12. In such settings, we are able to illustrate the effectiveness of our approach with practical examples for each inheritance relationship. As shown in Table 15, the relationship between “*think-PRON-AUX*”, “*believe-PRON-AUX*” and “*know-PRON-AUX*” belongs to polysemous inheritance. They can all be considered as instantiations of construction “*VERB-PRON-AUX*”. Thus, the subjective belief that someone has done or can do something is the primary meaning of this type of constructions. After that, different verbs are filled in the slots to inject expanded meanings. As for subpart inheritance, “*SCONJ-PRON-AUX*” is the smallest independent construction in the example. It is embodied in “*SCONJ-PRON-AUX-VERB*”, while “*SCONJ-PRON-AUX-VERB*” is also a subpart of “*SCONJ-PRON-AUX-VERB-to*”. The instance inheritance establishes the con-

Model	Rest14	Lap14	Rest15	Rest16
BERT-SPC	84.91	77.59	81.55	90.26
R-GAT	85.54	78.53	84.13	90.90
KumaGCN	85.09	79.00	83.76	91.07
DGEDT	85.27	78.67	82.84	90.42
DualGCN	85.44	78.84	83.95	91.23
HyCxG	85.98	79.31	84.32	91.88

Table 16: Experimental results on aspect-based sentiment analysis datasets with validation set.

nection between the construction and its completely or partially filled constructions. “*if-PRON-can*” is a specific construction that fills “*if*” and “*can*” into “*SCONJ-PRON-AUX*” as a substitute for “*SCONJ*” and “*AUX*”. Besides, it is worth mentioning that I_P is undirected relationship, while I_S and I_I are directed relationships in our *Constructicon*.

Furthermore, we develop a visualization toolkit to demonstrate the network of constructions. As shown in Figure 5, a sub-network that depicts the three inheritance relationships is constructed as an example under our settings.

K Detailed Results on ABSA Tasks

Since all four ABSA datasets (Rest14, Lap14, Rest15 and Rest16) of SemEval do not have official validation sets, randomness may be introduced for performance comparison. Therefore, we randomly divide the training set into a new training set and a validation set with the ratio of 9:1 for these ABSA datasets. Then we conduct the experiments with baseline models and our HyCxG. Meanwhile, the hyper-parameter searching is conducted for each model for fair comparison.

As shown in Table 16, we can observe that the performance comparison of the model is almost consistent with that in Table 2. First, the baseline models that inject syntactic information always achieve better performances than BERT-SPC. Second, R-GAT outperforms the other baseline models on the Rest14 and Rest15 datasets, while KumaGCN and DualGCN have better performance on Lap14 and Rest16, respectively. Third, HyCxG outperforms all baseline models with constructional information incorporated on these datasets, which illustrates the validity of HyCxG. Meanwhile, it can also substantiate that construction grammar can be regarded as the inductive bias to enhance the language representation in NLU tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Limitations are presented in a standalone Limitation section (after the Conclusion and before the References).
- A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Main claims are stated in Abstract and Introduction (Section 1).
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Scientific artifacts are illustrated in Section 2 and 3.

- B1. Did you cite the creators of artifacts you used?
Citations are all shown in the References and footnotes.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The information of artifacts is shown in Section B of the Appendix.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
The statistics are included in Section B of the Appendix.

C Did you run computational experiments?

The computational experiments are included in Section 4 and Section D, F, G, H, I, K in the Appendix.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The computational statistics are illustrated in Section D of the Appendix.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

The experimental setup is shown in Section 4.1 and Section E of the Appendix.

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

The descriptive statistics are included in Section 4.

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

The existing packages in our work are described in Section 4 and Appendix F.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.