Sequence Labeling as Non-Autoregressive Dual-Query Set Generation

Xiang Chen, Lei Li, Yuqi Zhu, Shumin Deng, *Member, IEEE*, Chuanqi Tan, Fei Huang, Luo Si, Ningyu Zhang, *Member, IEEE*, and Huajun Chen, *Member, IEEE*

***Abstract*—Sequence labeling is a crucial task in the NLP community that aims at identifying and assigning spans within the input sentence. It has wide applications in various fields such as information extraction, dialogue system, and sentiment analysis. However, previously proposed span-based or sequence-to-sequence models conduct locating and assigning in order, resulting in problems of error propagation and unnecessary training loss, respectively. This paper addresses the problem by reformulating the sequence labeling as a non-autoregressive set generation to realize locating and assigning in parallel. Herein, we propose a Dual-Query Set Generation (`DQSetGen`) model for unified sequence labeling tasks. Specifically, the dual-query set, including a prompted type query and a positional query with anchor span, is fed into the non-autoregressive decoder to probe the spans which correspond to the positional query and have similar patterns with the type query. By avoiding the autoregressive nature of previous approaches, our method significantly improves efficiency and reduces error propagation. Experimental results illustrate that our approach can obtain superior performance on 5 sub-tasks across 11 benchmark datasets. The non-autoregressive nature of our method allows for parallel computation, achieving faster inference speed than compared baselines. In conclusion, our proposed non-autoregressive dual-query set generation method offers a more efficient and accurate approach to sequence labeling tasks in NLP. Its advantages in terms of performance and efficiency make it a promising solution for various applications in data mining and other related fields.**

***Index Terms*—Sequence labeling, non-autogressive, set generation, transformer.**

Manuscript received 28 April 2023; revised 17 November 2023 and 6 January 2024; accepted 11 January 2024. Date of publication 5 February 2024; date of current version 26 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62206246; in part by the Fundamental Research Funds for the Central Universities under Grant 226-2023-00138; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LGG22F030011; in part by Ningbo Natural Science Foundation under Grant 2021J190; in part by CAAI-Huawei MindSpore Open Fund, Yongjiang Talent Introduction Programme, under Grant 2021A-156-G; in part by the CCF-Baidu Open Fund, Information Technology Center and State Key Lab of CAD&CG, Zhejiang University; and in part by NUS-NCS Joint Laboratory under Grant A-0008542-00-00. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zoraida Callejas. *(Xiang Chen and Lei Li are co-first authors.) (Corresponding author: Ningyu Zhang.)*

Xiang Chen, Lei Li, Yuqi Zhu, Ningyu Zhang, and Huajun Chen are with the Donghai Laboratory, Zhejiang University, Hangzhou 310058, China (e-mail: xiang_chen@zju.edu.cn; leili21@zju.edu.cn; zhuyuqi@zju.edu.cn; zhangningyu@zju.edu.cn; huajunsir@zju.edu.cn).

Shumin Deng is with NUS-NCS Joint Lab, National University of Singapore, Singapore 119077 (e-mail: shumin@nus.edu.sg).

Chuanqi Tan, Fei Huang, and Luo Si are with Alibaba Group, Hangzhou 311121, China (e-mail: chuanqi.tcq@alibaba-inc.com; f.huang@alibaba-inc.com; luo.si@alibaba-inc.com).

The code is available at https://github.com/zjunlp/DQSetGen.

Digital Object Identifier 10.1109/TASLP.2024.3358053

I. Introduction

SEQUENCE labeling (SL) is a crucial technique in natural language processing (NLP) that involves the identification of segment boundaries and their associated categories within texts. SL has a broad range of applications in NLP, including named entity recognition (NER) [1], [2], [3], [4], slot filling [5], and POS tagging [6]. This field of research has enabled the development of a wide variety of downstream tasks, including relation extraction, intent detection, and more. For instance, recent studies have shown that SL can be leveraged for reasoning-based relation extraction [7], [8], as well as for more complex tasks like intent detection with graph-based neural networks such as the Graph Interaction Network (GL-GIN) and the DialoGLUE dataset [9], [10]. The advancements in SL research have greatly contributed to the development of NLP technologies [11], and hold significant potential for further innovation in the future.

Previous works in sequence labeling have primarily employed labeling-based methods that assign a single label to each token in a sentence [12]. However, these methods are limited in their ability to identify overlapped spans, such as nested entities, as tokens in these cases may be associated with multiple labels, as shown in Fig. 1. Span-based methods have been developed to reconcile labeling-based models [13], [14], [15]. These methods enumerate all possible spans and classify them through various approaches instead of directly assigning labels to each token. Despite their effectiveness, span-based methods have some drawbacks, such as error propagation due to incorrect boundary identification, high computational costs, and ambiguity when applied to discontinuous tasks. To address these issues and enable the detection of both nested and discontinuous spans, researchers have proposed framing sequence labeling as Seq2Seq generation [16], [17], [18]. In this approach, the output is serialized into text, and expressive black-box neural networks are used to predict the flattened string encoding the spans. However, the labels in sequence labeling tasks are not in a specific order, and the one-by-one decoding strategy of Seq2Seq may result in unnecessary training loss when the model predicts the label correctly but in a different order than that defined.

To address the above issues, we first revisit the sequence labeling task from a unified perspective. Fundamentally, all SL tasks can be modeled as two atomic transformation operations:

1) *Locating*, which locates the position of the desirable spans [19].

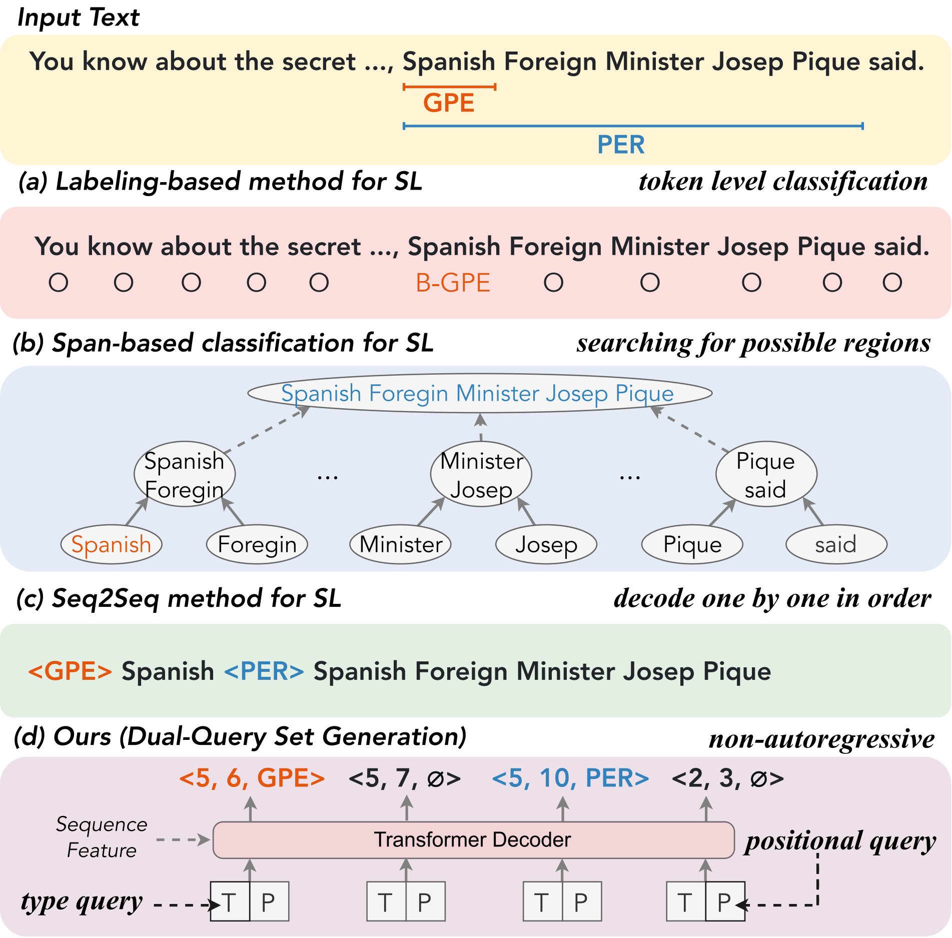
Input Text
You know about the secret ..., Spanish Foreign Minister Josep Pique said.
GPE
PER
(a) Labeling-based method for SL
token level classification
You know about the secret ..., Spanish Foreign Minister Josep Pique said.
O O O O O B-GPE O O O O O
(b) Span-based classification for SL
searching for possible regions
Spanish Foregin Minister Josep Pique
Spanish Foregin
...
Minister Josep
...
Pique said
Spanish
Foregin
Minister
Josep
Pique
said
(c) Seq2Seq method for SL
decode one by one in order
<GPE> Spanish <PER> Spanish Foreign Minister Josep Pique
(d) Ours (Dual-Query Set Generation)
non-autoregressive
<5, 6, GPE> <5, 7, ∅> <5, 10, PER> <2, 3, ∅>
Sequence Feature
Transformer Decoder
positional query
type query
T P
T P
T P
T P

Fig. 1. Comparison of various Sequence Labeling methods. **DQSetGen** conduct atomic transformation operations: 1) **Locating** and 2) **Assigning** in parallel.

2) *Assigning*, which assigns spans with specific semantic types in pre-defined schemas.

Traditionally, span-based methods have addressed these two atomic operations separately in two steps, while Seq2Seq methods have decomposed them into text-to-structure one by one in order. Based on the above fundamental observation of the SL task, we consider combining these two operations into one step with a non-autoregressive set generation framework, where learnable query vectors are assigned to spans, and multiple instance sets are fed into a non-autoregressive decoder to be decoded in parallel. This approach is designed to enhance the efficiency and accuracy of sequence labeling tasks by reducing the number of decoding steps required.

However, the set generation method still faces limitations in terms of learnable query sets: 1) *Without Semantics for Assigning* 2) *Without Position Prior for Locating*. We then propose *Dual-Query Set Generation* (**DQSetGen**) to tackle these problems. Given an input text, we extract sequence features with BERT [20]. Then the dual-query set, including a type query (semantic embedding) and a positional query (anchor spans), is fed into the non-autoregressive decoder to identify text spans that align with the positional criteria and exhibit semantic congruency with the type query. The iterative refinement of the dual-query set occurs progressively through each layer of the model, converging on the precise target spans. Decoder layer outputs serve the purpose of simultaneously deducing the labeled entities and their respective anchors, employing a single-pass inference strategy. To compute the loss during training, a bipartite graph matching approach is utilized, ensuring optimal alignment between the predicted and ground truth entities. Our proposed approach overcomes the limitations of the set generation method by incorporating both semantics and positional priors. This allows us to improve the accuracy of sequence labeling tasks while maintaining efficiency. We summarize the primary contribution of this paper as:

- We introduce a non-autoregressive set generation framework with dual queries for unified sequence labeling, including not only slot filling, POS tagging and nested NER but also discontinuous NER with more complex structures. To the extent of our knowledge, we are the first to employ dual queries that integrate type-related semantics and position priors to form trainable query sets, which results in exceptional performance and interpretability.
- Since we infer a fixed set of queries with a non-autoregressive decoder in parallel, thus, our framework is not as sensitive to the label order as the Seq2Seq model and avoids as time-consuming as span-based methods that search for all possible spans. We believe that our approach

has significant potential to enhance the performance of sequence labeling tasks and can be applied to a wide range of applications.

- Extensive experiments illustrate that our **DQSetGen** achieves exceptional performance across five sub-tasks, comprising a diverse range of Named Entity Recognition (NER), slot filling, and POS tagging tasks. Additionally, our model also features a significantly faster inference speed, being approximately 3.8 times faster than span-based models and 6.2 times quicker than Seq2Seq approaches on the ACE 2005 dataset.

II. Related Work

A. Sequence Labeling

Sequence labeling [21], [22], [23], [24] is a critical task in natural language processing (NLP) that can be used for a variety of applications, including slot filling, part-of-speech (POS) tagging and named entity recognition (NER). Recent advances in sequence labeling methods have leveraged pre-trained language models (PLMs) [25], [26], [27] such as RoBERTa [28], BART [29], and T5 [30] to perform token-level classification using powerful text encoding capabilities. Span-based classification methods [31], [32], [33], [34] have also been proposed to extend token-level classification to span-level classification. In addition to these classification-based methods, there have been efforts to use sequence-to-sequence (Seq2Seq) PLMs to handle sequence labeling tasks [17], [35], [36]. These models generate target spans and their tag labels autoregressively through pointer-based indexing [16] or tagging mechanisms [37]. Researchers have also focused on decoding strategies. For instance, Zhao et al. [38] proposes a novel hierarchical decoding model that dynamically parses act, slot, and value in a structured manner.

In this work, we propose a novel approach to sequence labeling that differs from both span-based and Seq2Seq methods. We model various sequence labeling tasks as dual-query set generation in a non-autoregressive manner, which eliminates error propagation and enables us to tag spans in one step. Specifically, our proposed method leverages a dual-query set, including a type query (semantic embedding) and a positional query (anchor spans), to probe the input text and generate sets of spans with specific semantic types. By modeling these tasks as the dual-query set generation, we can eliminate error propagation and tag spans in one step, which can improve performance while maintaining efficiency.

B. Set Generation

The set generation has been explored in machine learning, where the output is a set of unordered labels. In the field of computer vision, researchers have investigated set generation approaches for Transformer-based object detection models, such as DETR [39] that uses learnable query sets for task-specific features. However, the influence of order on the performance of various NLP tasks has been highlighted in previous studies, including graph generation [40], NER [41], keyphrase generation [42], and multi-label classification [43]. Other approaches have been proposed to explicitly model set properties for these tasks, such as conducting an exhaustive search for the suitable order [44] or modifying the optimization of the model [45]. Among the above-described related work about set generation, Tan et al. [41] introduces an innovative sequence-to-set model for nested NER that utilizes a set of fixed, trainable vectors to learn valuable span patterns, which is closely related to our **DQSetGen**. However, Tan et al. [41] concatenates the BERT contextualized embeddings, the GloVE embeddings, part-of-speech (POS) embeddings and character-level embeddings together, which runs counter to our simple application of BERT embedding for sequence labeling tasks. Thus, we don't compare with Tan et al. [41] in terms of NER sub-tasks. In contrast, we propose a dual-query set generation framework in a non-autoregressive manner that enables the model to identify and assign spans quickly, which is a fundamental requirement of sequence labeling.

III. Methodology

To construct the backbone $\mathcal{L} = [\mathcal{L}_{enc}, \mathcal{L}_{dec}]$ of the set generation, we adopt BERT [20] as the **Encoder** $\mathcal{L}_{enc}$, and a standard three-layer Transformer architecture as the **Decoder** $\mathcal{L}_{dec}$. In the following sections, we will provide a more detailed explanation of each component.

A. Task Formulation

Sequence labeling is a widely used task in natural language processing, which aims to predict a sequence of labels that correspond to a given input sequence. In this work, we approach sequence labeling as a non-autoregressive set generation problem, where each label is represented as a set of triples. Specifically, given a training sample (X, Y), where $X = [w_1, w_2, \ldots, w_n]$ is the input sequence of length n, and Y is the set of corresponding golden sequence labels. $Y = \{[Y_{k,s}^{(l)}, Y_{k,s}^{(r)}]_{s=1}^{S}, Y_k^t\}_{k=1}^{G}$, where $Y_{k,s}^{(l)}$ and $Y_{k,s}^{(r)}$ denote the left and right boundary positions of the s-th boundary of the k-th target span, Y_k^t is its corresponding label type, and G is the number of target spans. We use $\mathcal{E}$ to refer to the set of possible label types. In some cases, such as discontinuous entity recognition, S can be greater than 1. To enable non-autoregressive set generation, we assign U dual-query sets for each training sample, where U is greater than G. We then propose a four-component model that comprises sequence feature encoding, dual-query set construction, boundary-aware decoding, and learning with bipartite matching, using a BERT encoder and a standard three-layer Transformer decoder as the backbone. Details of the model are provided as follows.

B. Sequence Feature Encoding

Sequence labeling tasks require the model to capture not only the semantic meaning of the tokens but also their sequential nature. To achieve this, we use a BERT-based architecture as our encoder $\mathcal{L}_{enc}$ to obtain the final contextualized *sequence semantic feature* $\mathbf{X}_s = \{x_1, x_2, \ldots, x_n\}$ as:

$$\mathbf{X}_s = \mathcal{L}_{enc}(X) \tag{1}$$

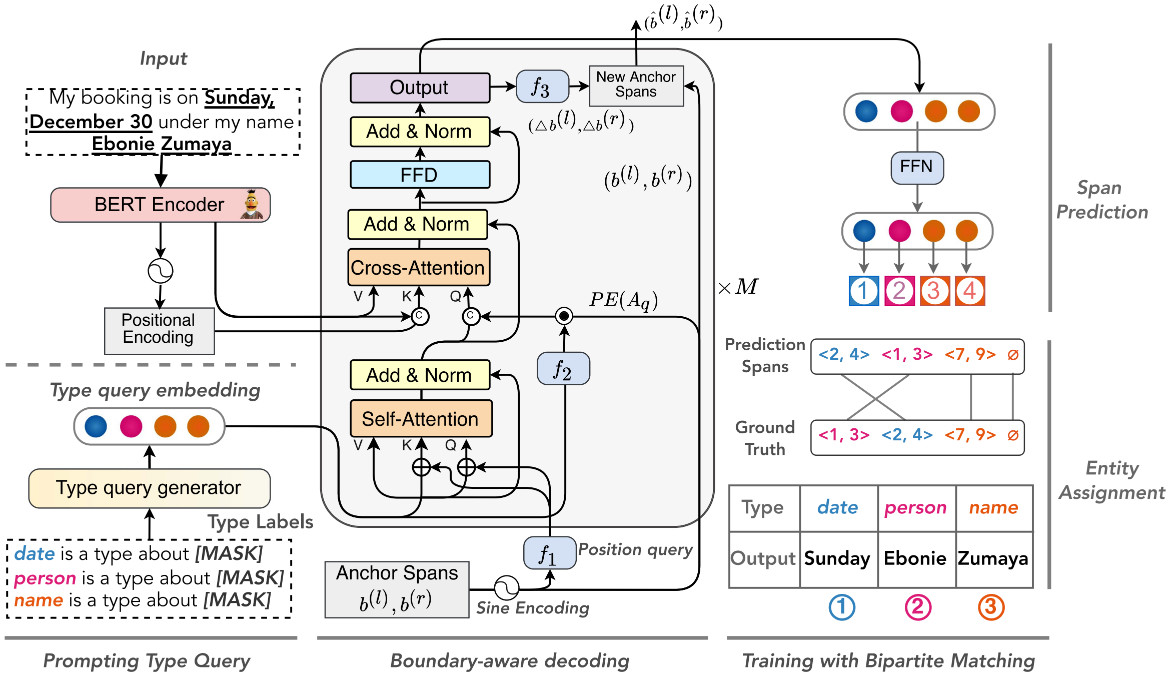

Fig. 2. Illustration of our **DQSetGen**. The type query embeddings are further updated in the following learning. It is important to highlight that the prediction heads (f_1, f_2, f_3) are implemented with shared parameters.

where n is the length of the input sequence $X = [w_1, w_2, \ldots, w_n]$.The dimension of vector of $\mathcal{L}$ is d, and therefore, $\mathbf{X}_s \in \mathbb{R}^{n\times d}$.

To incorporate the positional information of the tokens, we adopt a sinusoidal positional encoding method similar to that used in the original Transformer model [46]. Specifically, we define the sine encoding function PE$(\cdot)$ to represent the position embedding of the t-th token as follows:

$$\mathrm{PE}_{t,2i} = \sin\left(t/10000^{2i/d}\right),$$
$$\mathrm{PE}_{t,2i+1} = \cos\left(t/10000^{2i/d}\right), \tag{2}$$

The primary objective of the $PE(\cdot)$ function is to exclusively represent positional information. We then apply PE$(\cdot)$ to the sequence semantic feature $\mathbf{X}_s$ to obtain the *sequence positional feature* $\mathbf{X}_p \in \mathbb{R}^{n\times d}$, which takes into account the position of each token in the sequence. We denote $\mathbf{X}_p$ as:

$$\mathbf{X}_p = \mathrm{PE}\left(\mathbf{X}_s\right) \tag{3}$$

As illustrated in Fig. 2, both X_p and X_s interact within the cross-attention module of the decoder, enabling the incorporation of positional information through their mutual interaction.

C. Dual-Query Set Construction.

1) Prompting Type Query With Semantics: To enhance our model's ability to capture the semantic information of different types, we propose a type query generator that generates prompts for each sequence type $e \in \mathcal{E}$. The prompts are generated by filling in the corresponding type in a predefined template. As shown in the Fig. 2, for the sequence type e of *person*, the prompt $Pt^{(e)}$ could be formulated as "$\underline{person}isatypeabout\underline{[MASK]}$". Then, we use a vanilla BERT model as the type query generator and feed the prompt $Pt^{(e)}$ into it. The type query generator remains unchangeable in the following steps. The embedding $T_q \in \mathbb{R}^d$ of the q-th prompted type query is calculated as the output embedding of the "[MASK]" position:

$$T_q = \mathrm{BERT}\left(Pt^{(e)}\right)_{\texttt{[MASK]}} \tag{4}$$

This design enables the type queries to be initialized with the semantic information of different pre-defined types. The selected patterns for the type query generator in this study were intentionally kept simple, as illustrated in the paper. This choice was made to ensure a consistent pattern for comparison with baseline models. It is important to note that the simplicity of the prompt patterns does not undermine the effectiveness of our proposed method, as demonstrated in the empirical validation.

2) Positional Query With Anchor Span: To enhance the accuracy of dual-query set, we introduce the q-th anchor, denoted as $A_q = (b_q^{(l)}, b_q^{(r)})$, which refers to the q-th possible span associated with the q-th type query. The anchor span $A_q = (b_q^{(l)}, b_q^{(r)})$ refers to the q-th possible span associated with the q-th type query. Note that $b_q^{(l)}$ and $b_q^{(r)}$ are floating-point values within the range of [0,1] that indicate the left and right boundaries of the corresponding span, respectively, rather than vectors. The left and right boundaries, $(b_q^{(l)}, b_q^{(r)})$, of the anchor span, are generated from a uniform distribution over the interval [0, 1], subject to the constraint that $b_q^{(l)} < b_q^{(r)}$. We also define $P_q \in \mathbb{R}^d$ as corresponding positional query. The positional query P_q is generated using the positional encoding function PE$(\cdot)$, which maps the float values of $b_q^{(l)}$ and $b_q^{(r)}$ to a vector in $\mathbb{R}^{d/2}$. Given an anchor A_q of the q-th dual-query set, its positional query P_q

is generated by:

$$\mathrm{PE}(A_q) = \mathrm{PE}\left(b_q^{(l)}, b_q^{(r)}\right) = \left[\left(\mathrm{PE}\left(b_q^{(l)}\right); \mathrm{PE}\left(b_q^{(r)}\right)\right], \quad (5)$$

$$P_q = f_1\left(\mathrm{PE}(A_q)\right), \quad (6)$$

where the notion [;] means concatenation function, PE denotes positional encoding in (2) to generate sinusoidal embeddings from float numbers. Here we utilize the positional encoding function PE$(\cdot)$ mapping float value of $b_q^{(l)}, b_q^{(r)}$ to a vector as: PE: $\mathbb{R} \rightarrow \mathbb{R}^{d/2}$. The function $f_1(\cdot)$ is assigned as the module with a linear MLP mapping ($\mathbb{R}^d \rightarrow \mathbb{R}^d$) layer and a ReLU activation. The use of positional queries allows us to take the relative positions of the entities and relations into account, improving the positioning ability of the dual-query set.

D. Boundary-Aware Decoding

1) Dual-Query Across Attention: In our transformer-based decoder, each layer consists of two attention modules: a self-attention module and a cross-attention module. These modules serve different purposes: the self-attention module is used to update the dual-query sets, while the cross-attention module explores sequence features. To perform the attention operation, we use queries, keys, and values, which are different depending on the module used. For self-attention, we assign queries, keys, and values with the same type of items and supplement queries and keys with additional positional items. Specifically, we concatenate the type query and positional query as queries and use the sequence semantic features and sequence positional features as keys:

$$\text{Self-Attn: } Q_q = T_q + P_q, K_q = T_q + P_q, V_q = T_q, \quad (7)$$

Here, Q_q, K_q, and V_q denote queries, keys, and values, respectively. T_q represents the type query and P_q represents the positional query, which is generated based on 1-D coordinates using the positional encoding function PE$(\cdot)$.

The cross-attention module is designed to decouple the contributions of type and position in determining query-to-feature similarity, which is computed as the dot product between a query and a key. In this module, we concatenate the type and positional queries to form the overall query. Similarly, the sequence semantic features and sequence positional features are combined to create the keys. This approach allows for a more nuanced and precise interpretation of the interplay between type and position in the similarity assessment process. We enhance the process by incorporating an additional function $f_2(\cdot)$, which is learned through a MLP layer coupled with a ReLU activation. This function is specifically designed to generate a scale vector that is contingent on content information. Subsequently, this scale vector is employed to execute an element-wise multiplication with the positional embeddings, thereby enabling a more content-sensitive adjustment of these embeddings. The cross-attention operation is defined as follows:

$$\text{Cross-Attn: } Q_q = \left[F_{self}; \mathrm{PE}(A_q) \cdot f_2(T_q)\right],$$
$$K_q = \left[\mathbf{X}_s; \mathbf{X}_p\right], \quad V_q = \mathbf{X}_s, \quad (8)$$

where $\cdot$ is an element-wise multiplication, F_{self} is the output of the self-attention module,$\mathbf{X}_s$ and $\mathbf{X}_p$ represent the sequence semantic features and sequence positional features, respectively.

2) Modulating Anchor Span Layer-By-Layer.: The accuracy of initial anchor coordinates is crucial to the success of learning, as they are the basis for updating anchor spans layer by layer. However, in many cases, the initial anchor coordinates are not accurate and have no direct correlation with the sequence content. To address this issue, we propose to use coordinates as queries for learning, enabling the updating of anchor spans layer by layer through a prediction head $f_3(\cdot)$. As illustrated in Fig. 2, the prediction head consists of multiple MLP layers and ReLU activation functions, which predict relative positions $(\Delta b_q^{(l)}, \Delta b_q^{(r)})$ based on the output of the cross-attention mechanism of each layer. The updated anchor spans in each layer are then obtained by modulating the initial anchor spans with the predicted relative positions as follows:

$$\left(\Delta b_q^{(l)}, \Delta b_q^{(r)}\right) = f_3\left(F_{\text{cross}}\right), \quad (9)$$

$$b_q^{\hat{(l)}} = \sigma\left(\sigma^{-1}\left(b_q^{(l)}\right) + \Delta b_q^{(l)}\right)$$
$$b_q^{\hat{(r)}} = \sigma\left(\sigma^{-1}\left(b_q^{(r)}\right) + \Delta b_q^{(r)}\right), \quad (10)$$

where F_{cross} is the output of the cross-attention mechanism of each layer, σ is the sigmoid function, and σ^{-1} denotes its inverse function, which maps the output to the real number space. The updated anchor spans are therefore obtained by adding the predicted relative positions to the initial anchor spans after being transformed by the sigmoid function. Our proposed method can effectively update the initial anchor spans layer by layer and achieve better performance compared to traditional methods.

E. Training With Bipartite Matching

1) Predictive Head: The FFN module in Fig. 2 is composed of several layers of MLP, where the output of the final layer is passed through the Softmax function to predict the probability of classification for types. We denote the final output embeddings from the decoder as $\mathrm{H} \in \mathbb{R}^{U \times d}$, where U is the number of target spans. Since we predict a fixed-size set of U spans, we assign an additional label $\varnothing$ to signify that no target span has been identified.

Given the set embedding $\mathrm{h} \in \mathbb{R}^d$ in H, we calculate the set generation process as follows. First, we use the Softmax function and an MLP layer to calculate the probability p^c of classification for types. Then, we duplicate h for n times into shape $\mathbb{R}^{n \times d}$ using the function dup. We concatenate $\mathrm{X_s}$ with the duplicated h to generate a new embedding $\hat{\mathrm{h}}$. Finally, we use Softmax and two MLP layers to calculate the probabilities $p_s^{(l)}$ and $p_s^{(r)}$ for the left and right boundaries of the s-th sub-span. The specific set generation process is calculated as:

$$p^c = \mathrm{Softmax}\left(\mathrm{MLP}_c(\mathrm{h})\right), \quad (11)$$

$$\hat{\mathrm{h}} = \left[\mathrm{dup}(\mathrm{h}, n); \mathrm{X_s}\right], \quad (12)$$

$$p_s^{(l)} = \mathrm{Softmax}\left(\mathrm{MLP}_s^{(l)}\left(\hat{\mathrm{h}}\right)\right), \quad (13)$$

$$p_s^{(r)} = \text{Softmax}\left(\text{MLP}_s^{(r)}\left(\hat{\text{h}}\right)\right), \tag{14}$$

Note that the enactment of s-th sub-span is specifically for discontinuous NER, where we assign several additional dual-query sets for the discontinuous entities with different numbers of sub-spans. Overall, our proposed method utilizes the predictive head module to predict the probabilities of classification for types and the boundaries of sub-spans, which enables us to accurately identify both continuous and discontinuous entities.

2) Bipartite Matching Loss: To train the model effectively, our objective is to evaluate the predicted spans, characterized by their *left*, *right*, and *class* attributes, against the reference gold spans. We denote the gold standard set by y, and the collection of U predictions is represented as $\hat{y} = \{\hat{y}_i\}_{i=1}^{U}$. We further pad y to the size of U with $\varnothing$ for which having no specific types to the golden answer set. Moreover, we optimize the bipartite matching loss calculation by efficiently sampling optimally matched gold spans. We search for a permutation of U elements $\alpha \in \mathcal{Z}_U$ with the lowest cost and leverage the classical Hungarian algorithm [47] for efficient optimal assignment as:

$$\hat{\alpha} = \underset{\alpha \in \mathcal{Z}_U}{\arg\min} \sum_{i}^{U} \mathcal{J}_{\text{m}}\left(y_i, \hat{y}_{\alpha(i)}\right), \tag{15}$$

$$\begin{aligned}\mathcal{J}_{\text{m}}\left(y_i, \hat{y}_{\alpha(i)}\right) &= -\mathbb{1}_{\{c_i \neq \varnothing\}}\left[p_{\alpha(i)}^{c}\left(c_i\right)\right.\\ &\qquad \left.+\, p_{\alpha(i)}^{(l)}\left(l_i\right) + p_{\alpha(i)}^{(r)}\left(r_i\right)\right],\end{aligned} \tag{16}$$

Note that each element i of the golden set can be regarded as a $y_i = (l_i, r_i, c_i)$. Upon finding the optimal assignment $\hat{\alpha}(i)$, we define the final training loss over the predicted span in $\hat{y}$ and the ground truth span in y, and regarding the output as:

$$\begin{aligned}\mathcal{J}(y, \hat{y}) = \sum_{i=1}^{N} &\left\{-\log p_{\hat{\alpha}(i)}^{c}\left(c_i\right)\right.\\ &+ \mathbb{1}_{\{c_i \neq \varnothing\}}\left[-\log p_{\hat{\alpha}(i)}^{(l)}\left(l_i\right)\right.\\ &\left.\left.-\log p_{\hat{\alpha}(i)}^{(r)}\left(r_i\right)\right]\right\}.\end{aligned} \tag{17}$$

IV. Experiments

A. Experimental Settings

1) Datasets: We evaluate our proposed model on a variety of well-established sequence labeling tasks from multiple domains. Specifically, The flat NER dataset we use is OntoNotes [48], while the two discontinuous NER datasets are ShARe13 [49] and ShARe14 [50]. For nested NER, we use three datasets, namely KBP17 [51], ACE04 [52], and ACE05 [53]. We also evaluate our model on two widely used POS datasets: Wall Street Journal (WSJ) [54] and CoNLL 2003 [55]. In addition, we conduct experiments on three slot filling datasets from DialoGLUE [10], including RESTAURANT8K [56], MixATIS [57], [58], and MixSNIPS [59].

2) Implementation Details: We follow standard evaluation metrics used in previous works. To ensure a fair comparison with existing approaches, we utilize the BERT [20] as our encoder, and a 3-layer Transformer as our decoder module. Specifically, we use BERT-large on the named entity recognition (NER) dataset and BERT-base on the slot filling and part-of-speech (POS) dataset. We initialize the weights in the encoder architecture using pre-trained BERT models and the decoder architecture with a 3-layer Transformer. Training is conducted on a single NVIDIA-V100 GPU. We train models with 3 fixed seeds and several learning rates [1e-5, 2e-5, 3e-5]. The final reported performance is the average of results across the 3 different seeds. We use the Adam optimizer with a learning rate warmup strategy and a linear decay schedule for optimization. Additionally, we apply early stopping and conduct model selection based on the performance of the validation.

B. Compared Baselines

We compare our **DQSetGen** with previous labeling-based approaches, span-based approaches and sequence-to-sequence methods, respectively. As for labeling-based approaches, we adopt Straková et al. [60], Shibuya and Hovy [36] as baselines for NER tasks, BERT-CRF and Wang et al. [32] as baselines for slot filling and pos tagging tasks. We further assign Luan et al. [61], Li et al. [14], Yu et al. [15], Wang et al. [62] as the span-based baselines for NER tasks, and Joshi et al. [63], Jiang et al. [64] as span-based baselines for slot filling and POS tagging tasks.

Apart from previous approaches, we compare our proposed method against several baseline models from the literature. Specifically, we consider the following baselines: Yan et al. [16], Zhang et al. [17], and Lu et al. [18], which primarily focus on NER tasks, and Athiwaratkun et al. [35], TANL [65], and GL-GIN [9], which are mainly involved in slot filling or POS tagging tasks. To facilitate a fair comparison, we present the results for each baseline in separate tables based on the type of task. Among these baselines, GL-GIN [9] stands out as a classical method for sequence labeling tasks. We consider it essential to include GL-GIN as one of the foundational baselines in our evaluation. Despite GL-GIN not utilizing the BERT architecture, we have included comparisons with other BERT-based approaches in our baselines to showcase the effectiveness of our proposed method. It is worth noting that UIE [18] can perform NER, slot filling, and POS tagging tasks, and we include it as a baseline model for comparison across all datasets.

C. Overall Performance

1) Performance on Named Entity Recognition: We evaluate our **DQSetGen** for named entity recognition (NER) on multiple datasets, comparing its performance against the main baselines. The results are presented in Table I, using F1-score, precision, and recall as the evaluation metrics. Our **DQSetGen** exhibits superior performance over the state-of-the-art (SOTA) model on the majority of datasets, underscoring its effectiveness. **DQSetGen** demonstrates notable improvements of +1.25% and +0.29% on the ACE 2004 and ShARe14 datasets, respectively. Additionally, it achieves comparable results with the state-of-the-art (SOTA) model on the ShARe13 dataset. Moreover, we explore the applicability of Seq2Seq models for complex tasks,

TABLE I
OVERALL PERFORMANCES FOR VARIOUS NER DATASETS

	Flat NER	**Nested NER**									**Discontinuous NER**					
	OntoNote	ACE2004			ACE2005			KBP17			ShARe13			ShARe14		
Model	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Labeling-based Approach																
Straková et al. (2019)	-	-	-	84.33	-	-	83.42	-	-	-	-	-	-	-	-	-
Shibuya and Hovy (2020)⋆	-	85.23	84.72	84.97	83.30	84.69	83.99	-	-	-	-	-	-	-	-	-
Span-based Approach																
Luan et al. (2019)[ELMO]	-	-	-	84.70	-	-	82.90	77.35	74.78	76.00	-	-	-	-	-	-
Li et al. (2020)⋆	89.84	85.83	85.77	85.80	85.01	84.13	84.57	82.33	77.61	80.97	-	-	-	-	-	-
Yu et al. (2020)⋆	89.83	85.42	85.92	85.67	84.50	84.72	84.61	81.85	78.93	80.36	-	-	-	-	-	-
Wang et al. (2020)⋆	-	86.08	86.48	86.28	83.95	85.39	84.74	82.32	79.85	81.07	-	-	-	-	-	-
Sequence-to-sequence Approach																
Yan et al. (2021)	89.76	87.27	86.41	86.84	83.16	86.38	84.74	**86.32**	84.04	85.16	82.07	76.45	79.16	75.88	**84.37**	79.90
Zhang et al. (2022)[T5-Base]	89.76	86.53	84.06	85.28	82.92	87.05	84.93	84.90	84.15	84.52	81.31	**76.75**	78.96	77.51	83.27	80.29
Lu et al. (2022)(SEL)"†"	89.88	86.26	86.68	86.47	84.02	86.29	85.14	85.22	**84.58**	84.90	-	-	-	-	-	-
DQSetGen[BERT-large+3dec]	**90.18**	**88.34**	**87.84**	**88.09**	**86.80**	**87.23**	**86.91**	86.26	84.48	**85.36**	**83.05**	75.89	**79.31**	**79.47**	81.73	**80.58**

We highlight the best result in bold. ⋆denotes the value from the experimental results listed in [16], "†" means our rerun of their code on the unreported datasets in their paper. As for Lu et al. [18], we adopt the UIE-large (SEL) without pre-training for a fair comparison.

TABLE II
COMPARISON ON SLOT FILLING AND POS DATASETS

	Slot Filling									**POS**					
	RESTAURANT8K			MixATIS			MixSNIP			CoNLL2003			WSJ		
Model	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Labeling-based Aapproach															
BERT-CRF	95.91	95.82	95.36	84.43	87.83	86.10	95.34	94.28	94.81	92.67	92.41	92.54	96.18	91.53	93.80
Wang et al. (2021)	94.88	95.89	95.38	82.55	88.31	85.33	95.78	96.17	95.97	93.68	93.05	93.36	96.83	96.23	96.53
Span-based Approach															
Joshi et al. (2020)	94.29	95.85	95.07	84.92	**88.34**	86.59	95.94	96.32	96.13	92.57	92.45	92.51	96.87	96.35	96.61
Jiang et al. (2020)	89.46	97.03	93.09	85.32	88.16	86.71	94.53	96.30	95.41	**94.47**	92.02	93.23	**97.24**	96.03	96.63
Sequence-to-sequence Approach															
Athiwaratkun et al. (2020)	95.35	96.13	95.74	87.21	85.43	86.31	95.55	94.12	94.83	91.25	91.99	91.62	96.34	95.45	95.89
Paolini et al. (2021)	96.55	97.38	96.96	86.22	86.95	86.58	95.78	95.48	95.63	93.78	92.15	92.96	96.55	92.74	94.61
GL-GIN Qin et al. (2021)	95.83	96.95	96.39	86.89	88.25	87.56	94.15	95.68	94.90	93.89	88.78	91.26	96.35	89.89	93.00
Lu et al. (2022)(SEL)	96.42	97.25	96.84	86.53	86.88	86.70	95.54	95.59	95.56	94.08	88.89	91.41	96.75	89.20	92.86
DQSetGen[BERT-base +3dec]	**96.76**	**97.70**	**97.23**	**88.02**	87.86	**87.94**	**96.49**	**96.58**	**96.54**	94.18	**93.14**	**93.66**	96.92	**97.19**	**97.05**

Considering most baselines utilize BERT-base in their paper, we reproduce all baselines with base LM for a fair comparison. "3dec" denotes 3 layers of the Transformer decoder.

such as discontinuous named entity recognition (NER). We compare them with span-based models and non-autoregressive set generation models. Our findings reveal that Seq2Seq models are more suitable for complex tasks than span-based models, but they are significantly outperformed by non-autoregressive set generation models. This observation suggests that non-autoregressive set generation models hold greater potential for handling complex tasks like discontinuous NER. In contrast to the Seq2Seq model, our proposed model effectively captures both semantic and positional information of spans by generating dual-query sets in parallel.

2) Performance on Slot Filling and Pos Tagging: In this section, we present the results of our proposed **DQSetGen** on slot filling and part-of-speech (POS) tagging tasks in addition to the NER task. We evaluate the overall performance of **DQSetGen** and baseline models on these tasks and report the results in Table II. The evaluation metrics used in our study are F1-score, precision, and recall. Our proposed **DQSetGen** outperforms several state-of-the-art (SOTA) span-based and sequence-to-sequence models on these tasks, achieving the best performance of all the datasets. Specifically, we achieve an improvement of +0.27%, +0.38%, and +0.98% on the RESTAURANT8, MixATIS, and MixSNIP datasets, respectively. These results demonstrate the effectiveness of our proposed approach for slot-filling tasks, which require identifying values for certain attributes or slots within the input sequence. Moreover, our proposed **DQSetGen** achieves comparable results with the state-of-the-art models on the POS tagging task, highlighting the robustness of our proposed approach on different tasks. This task involves assigning a part-of-speech tag to each word in the input sequence, which is an essential preprocessing step for many natural language processing (NLP) tasks.

D. Comparison of Inference Speed and Memory Usage

1) Inference Speed: In this section, we highlight the efficiency of our proposed **DQSetGen** compared to other state-of-the-art models. While the previous experiments have demonstrated the superiority of our method in terms of accuracy,

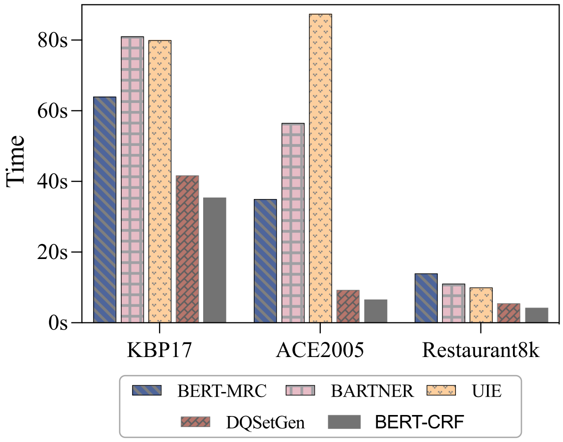

Fig. 3. Comparison of inference time on KBP17, ACE2005 and Restaurant8k. Results are obtained with single V100 GPU.

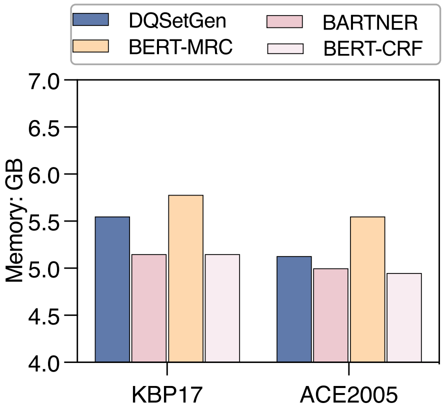

Fig. 4. Comparison of inference memory usage on KBP17 and ACE2005. Results are obtained with single V100 GPU.

our approach is also significantly faster in terms of inference speed. To demonstrate this, we provide a comprehensive comparison of the overall inference time between **DQSetGen** and other models, including the span-based model BERT-MRC, the Seq2Seq models BARTNER and UIE, and the labeling-based method BERT-CRF. The comparison is presented in Fig. 3. Our approach, **DQSetGen**, generates multiple sets simultaneously, making it typically 10x faster than UIE. In contrast, UIE follows a one-by-one paradigm to predict the structure language, resulting in slower performance. Other Seq2Seq models, such as BARTNER, which also have a sequential prediction structure and training process, are also likely to have similar speeds to UIE. Furthermore, our model performs even faster than BERT-MRC, which requires enumerating multiple question queries, incurring high computational costs.

2) Inference Memory Usage: Moreover, the GPU memory consumption during model inference is positively correlated with the size of its base model. Therefore, we also compared the GPU memory usage for inference in Fig. 4 to provide a comprehensive analysis. As the parameter sizes of BERT_large and BART_large are very similar, their inference memory usage is also comparable. Importantly, **DQSetGen** does not exhibit a significant increase in memory usage compared to labeling-based, span-based, and Seq2Seq-based methods.

E. Ablations Study

To ascertain the efficacy of **DQSetGen**, we execute a comprehensive ablation analysis. This analysis is meticulously designed to dissect the individual impact of the various components integrated within our framework. The insights from this investigation are delineated in Table III.

1) Prompted Type Query: We investigate the effectiveness of prompted type query, which is used to prompt the model to generate a certain type of entity. The results show that removing prompted type query leads to a clear drop in overall and classification F1-score on both MixSNIP and ACE04 datasets, decreasing classification F1-score by -0.66% and -0.44%, respectively. This indicates that prompted type query is a powerful technique that benefits type classification.

2) Positional Query: We examine the impact of the positional query, which is used to encode the positional information of the input text. Eliminating the positional query further decreases the location and overall F1 score on both MixSNIP and ACE04 datasets. Additionally, the elimination of positional query has a strongly negative effect on the discontinuous F1-score on the ShARe14 dataset. We speculate that positional query plays a critical role when locating spans with complicated structures in discontinuous NER. This suggests that positional query is crucial for improving the performance of **DQSetGen** in complex NER tasks.

3) Anchor Modulation: We investigate the effect of anchor modulation, which is used to capture implicit positional information through layer-by-layer updates. The model without anchor modulation achieves consistent drops on all three datasets, with an improvement over the model without positional query. This suggests that anchor modulation has the potential to capture implicit positional information, and its combination with a positional query can improve the performance.

4) Bipartite Matching: We analyze the effect of bipartite matching, which is used to ensure that the generated spans match the ground-truth spans. The exclusion of a bipartite matching loss results in an average decrease of 0.7% in the F1-score across three benchmark datasets. This is hypothesized to be due to the permutation-invariant property of bipartite matching, which is beneficial for the non-autoregressive generation. This indicates that bipartite matching is an important module for our proposed approach.

Overall, our ablation study demonstrates that each module in our **DQSetGen** plays a significant role in improving the performance of our model and provides insights into the design choices of our method.

F. Analysis

1) Visualization of Attention: To better understand the effectiveness of **DQSetGen**, we conduct a visualization of the attention map at the cross-attention computation, as shown in Fig. 5. From the visualization, we can observe that several

TABLE III
ABLATION STUDY

Model	MixSNIP			ACE04			ShARe14	
	Loc. F1	Cls. F1	F1	Loc. F1	Cls. F1	F1	F1	Dis. F1
Default	**98.03**	**97.52**	**96.54**	**91.04**	**91.34**	**88.09**	**80.58**	**52.24**
w/o Prompted Type Query	97.98	96.50	95.88	90.95	90.38	87.65	79.85	52.10
w/o Positional Query	96.22	97.40	95.25	88.28	91.20	85.88	79.36	51.38
w/o Anchor Modulation	96.60	97.38	95.45	88.98	91.23	86.27	79.41	51.55
w/o Bipartite Matching	97.42	96.00	95.55	89.27	90.75	87.33	79.53	51.79

(1) w/o Prompted Type Query, in which we replace the prompted type query with a randomly initialized embedding; (2) w/o Positional Query, in which we eliminate the whole position query and corresponding sequence positional embeddings; (3) w/o Anchor Modulation, in which we keep the dual-query set but eliminate the anchor modulation across layers; and (4) w/o Bipartite Matching, in which we replace the cross-entropy loss with bipartite matching loss.

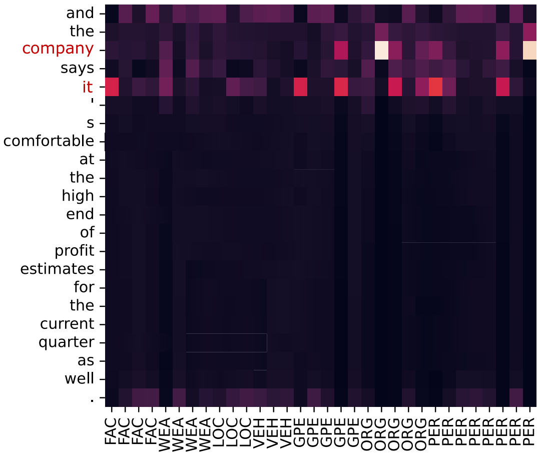

Fig. 5. Visualization of the cross-attention module. The x-axis denotes the type-specific dual-query sets, and the y-axis is allocated to represent the input sentence.

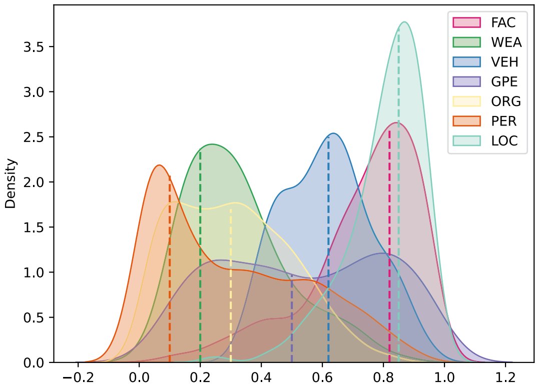

Fig. 6. Kernel density estimation of entity distribution. The dotted line indicates the central position of the anchor spans.

"ORG"-specific sets have the largest attention score on the target entity span "*company*", which demonstrates the effectiveness of our dual-query set. Moreover, we observe that different dual-query sets focus on spans at different positions, indicating that the instance sets can learn the query semantics related to span positions. This observation is consistent with our hypothesis that dual queries can provide more informative guidance for instance sets, enabling them to better capture the important features of target spans.

2) Impact of Modulated Anchor: We investigate the impact of our proposed modulated anchor on span location prediction. As shown in Fig. 6, we normalize the predicted central locations of the spans and use kernel density estimation to draw the distribution of the predicted span locations for different dual-query sets. We also draw the central locations of the modulated anchor to explore where it is located. The comparison illustrates that each set focuses on a different peak, and the average center position of the anchor is very close to the predicted span peak position, revealing the prior guidance of our anchor span on the location prediction. This finding further supports our hypothesis that modulated anchors can provide effective guidance for instance sets to better predict span locations.

3) Impact of Decoder Layers: We analyze to investigate the impact of the number of decoder layers on the performance of our proposed approach. As presented in Table V, we observe that when the number of decoder layers is reduced from 3 to 2 and from 2 to 1, the model's performance decreases by 0.53% and 0.93%, respectively. This result indicates that the model's depth plays an important role in capturing complex input features and generating accurate predictions. Furthermore, we also experiment with increasing the number of decoder layers to 6, but surprisingly, the proposed approach shows no significant improvement in performance. This result suggests that there is an optimal number of decoder layers for our proposed approach and that adding more layers does not necessarily lead to better performance. Our analysis demonstrates the insensitivity of the number of decoder layers. Despite this, potential limitations of our approach may lie in the need for multiple sets of hyperparameter tuning to search for the optimal decoder layer, which could potentially increase the overall training time.

4) Case Analysis: We perform a comprehensive case analysis comparing our model predictions to the golden labels to evaluate its performance, as shown in Table IV. **DQSetGen** demonstrates strong capabilities in accurately identifying and recognizing a wide range of spans, which can be attributed to the efficacy of our prompted type query. The majority of predicted span types align with the pre-defined type of query set, affirming the effectiveness of our approach in accurately assigning types. However, we also observe that our model's ability to understand sentences is still insufficient in certain cases, particularly in recognizing special phrases. For example, in case 1, the phrase "American tourists" is misclassified as FAC instead of the correct

TABLE IV
CASES ANALYSIS ON FOUR SEQUENCE LABELING TASKS

Task	Example with Gold Annotation	Generated Spans ← Prompted Type Query
Nested NER	Over the weekend, [^{4}[^{4}Islamic^{4}]$_{\text{PER}}$ militants^{4}]$_{\text{PER}}$ went to [^{8}hotels in [^{10}the central [^{13}Javanese^{13}]$_{\text{GPE}}$ city of [^{16}Solo^{16}]$_{\text{GPE}}$ ^{16}]$_{\text{GPE}}$ ^{16}]$_{\text{FAC}}$, demanding that [^{20}[^{20}American^{20}]$_{\text{GPE}}$ tourists^{21}]$_{\text{PER}}$ leave the country within 48 hours.	✓ (4, 4, PER) ← PER ✓ (10, 16, GPE) ← GPE ✓ ... ... ✗ (20, 21, FAC) ← FAC ✓ (20, 20, GPE) ← GPE
Discontinuous NER	Chest CT: [^{3}Multiple small^{4}]$_{\text{DIS}}$ less than 5 mm, [^{12}noncalcified pulmonary nodules^{14}]$_{\text{DIS}}$, which are likely of benign etiology, and a followup CT could be obtained in one year. [^{34}Calcified granuloma^{36}]$_{\text{DIS}}$ seen in the left upper lobe.	✓ (3, 12, 4, 14, DIS) ← DIS ✓ (34, 36, DIS) ← DIS
Slot Filling	We have a lunch reservation that we need to cancel for today iam afraid. It listed under [^{18}Josette^{18}]$_{\text{first\_name}}$ [^{19}Wilczynski^{19}]$_{\text{last\_name}}$ at [^{21}12:45^{21}]$_{\text{time}}$ for [^{23}10 people ^{24}]$_{\text{people}}$.	✓ (21, 21, time) ← time ✓ (18, 18, first_name) ← first_name ✓ (23, 24, people) ← people ✓ (19, 19, last_name) ← people
POS	But [^{1}Catholic^{1}]$_{\text{NNP}}$ church [^{3}officials^{3}]$_{\text{NNS}}$ said they had no [^{8}confirmation^{8}]$_{\text{NN}}$ [^{9}of^{9}]$_{\text{IN}}$ the report and would [^{14}hava^{14}]$_{\text{VB}}$ to wait until [^{18}Thursday^{18}]$_{\text{NNP}}$ to be [^{21}sure^{21}]$_{\text{JJ}}$.	✓ (9, 9, IN) ← IN ✓ (18, 18, NNP) ← DT ✓ ... ... ✗ (21, 21, JJR) ← JJR

The lower right label in the middle column indicates the entity type, while the superscripts denote the positions of the left and right boundary words. On the right column, we present the mapping between the prompted type queries and the generated spans.

TABLE V
ANALYSIS OF IMPACT ON DECODER LAYER NUMBER ON ACE2005 DATASET

# Layer	ACE 2005		
	P	R	F1
1	86.55	84.23	85.45
2	85.93	86.52	86.38
3	86.80	87.23	86.91
4	86.64	87.85	**87.24**
5	86.78	87.47	87.12
6	86.72	87.43	87.08

type PER. This suggests that improving the pre-training for common knowledge may be a future direction for enhancing our approach's understanding of special phrases.

V. CONCLUSION

In this paper, we propose a novel approach to sequence labeling tasks that employs non-autoregressive dual-query sets. By using a collection of dual-query sets, our method can probe input sequence features in parallel, allowing for the location and prediction of all target spans simultaneously. Our approach utilizes prompted type query and positional query to automatically learn semantic and positional information related to types and locations, enabling the model to locate and assign spans effectively. As large models are becoming more prevalent, the Seq2Seq method, which autoregressively decodes one by one in order, can be slow to infer. Our experimental results demonstrate that our non-autoregressive dual-query set generation achieves not only superior performance but also faster inference times than both span-based and Seq2Seq models. Furthermore, our **DQSetGen** approach can be easily extended to large-scale models and set pre-training in the future, making it a promising direction for future research in this area.

ACKNOWLEDGMENT

The authors would like to express gratitude to the anonymous reviewers for their kind comments.

REFERENCES

[1] T. Gui et al., "A lexicon-based graph neural network for chinese NER," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 1040–1050, doi: 10.18653/v1/D19-1096.

[2] X. Chen et al., "One model for all domains: Collaborative domain-prefix tuning for cross-domain NER," in *Proc. 32d Int. Joint Conf. Artif. Intell.*, 2023, pp. 5030–5038, doi: 10.24963/ijcai.2023/559.

[3] X. Wang et al., "InstructUIE: Multi-task instruction tuning for unified information extraction," 2023, *arXiv:2304.08085*.

[4] J. Wang et al., "TECHGPT-2.0: A large language model project to solve the task of knowledge graph construction," 2024, *arXiv:2401.04507*.

[5] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot filling models in natural language understanding," *ACM Comput. Surv.*, vol. 55, no. 8, 2023, Art. no. 156, doi: 10.1145/3547138.

[6] H. Zhou, Y. Li, Z. Li, and M. Zhang, "Bridging pre-trained language models and hand-crafted features for unsupervised POS tagging," in *Proc. Findings Assoc. Assoc. Linguistics:*, 2022, pp. 3276–3290, doi: 10.18653/v1/2022.findings-acl.259.

[7] G. Nan, Z. Guo, I. Sekulić, and W. Lu, "Reasoning with latent structure refinement for document-level relation extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 5-10, 2020, pp. 1546–1557. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.141

[8] X. Chen et al., "Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction," in *Proc. WWW: ACM Web Conf*, 2022, pp. 2778–2788, doi: 10.1145/3485447.3511998.

[9] L. Qin, F. Wei, T. Xie, X. Xu, W. Che, and T. Liu, "GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling," in *Proc. 59th Ann. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 178–188, doi: 10.18653/v1/2021.acl-long.15.

[10] S. Mehri, M. Eric, and D. Hakkani-Tür, "Dialoglue: A natural language understanding benchmark for task-oriented dialogue," *CoRR*, vol. abs/2009.13570, 2020. [Online]. Available: https://arxiv.org/abs/2009.13570

[11] S. Deng, S. Mao, N. Zhang, and B. Hooi, "SPEECH: Structured prediction with energy-based event-centric hyperspheres," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 351–363, doi: 10.18653/v1/2023.acl-long.21.

[12] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 260–270.

[13] B. Wang and W. Lu, "Combining spans into entities: A neural two-stage approach for recognizing discontiguous entities," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 6215–6223, doi: 10.18653/v1/D19-1644.

[14] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified MRC framework for named entity recognition," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5849–5859, doi: 10.18653/v1/2020.acl-main.519.

[15] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6470–6476, doi: 10.18653/v1/2020.acl-main.577.

[16] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various NER subtasks," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. on Natural Lang. Process.*, 2021, pp. 5808–5822. [Online]. Available: https://aclanthology.org/2021.acl-long.451

[17] S. Zhang, Y. Shen, Z. Tan, Y. Wu, and W. Lu, "De-bias for generative extraction in unified NER task," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 808–818, doi: 10.18653/v1/2022.acl-long.59.

[18] Y. Lu et al., "Unified structure generation for universal information extraction," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5755–5772, doi: 10.18653/v1/2022.acl-long.395.

[19] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 366–373, doi: 10.1109/CVPR.2004.77.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[21] H. Chen, Q. Ma, L. Yu, Z. Lin, and J. Yan, "Corpus-aware graph aggregation network for sequence labeling," *IEEE/ACM Trans. Audio, Speech Lang. Proc.*, vol. 29, pp. 2048–2057, 2021, doi: 10.1109/TASLP.2021.3084105.

[22] P. Zhu et al., "Improving chinese named entity recognition by large-scale syntactic dependency graph," *IEEE/ACM Trans. Audio, Speech Lang. Proc.*, vol. 30, pp. 979–991, 2022, doi: 10.1109/TASLP.2022.3153261.

[23] X. Chen et al., "LightNER: A lightweight tuning paradigm for low-resource NER via pluggable prompting," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 2374–2387. [Online]. Available: https://aclanthology.org/2022.coling-1.209

[24] X. Chen et al., "Continual multimodal knowledge graph construction," 2023, *arXiv.2305.08698*.

[25] N. Zhang et al., "A comprehensive study of knowledge editing for large language models," 2024, *arXiv:2401.01286*.

[26] W. X. Zhao et al., "A survey of large language models," 2023, doi: 10.48550/arXiv.2303.18223.

[27] Y. Yao et al., "Editing large language models: Problems, methods, and opportunities," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 10222–10240. [Online]. Available: https://aclanthology.org/2023.emnlp-main.632

[28] Y. Liu et al., "Roberta: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[29] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[30] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[31] H. Lin, Y. Lu, X. Han, and L. Sun, "Sequence-to-nuggets: Nested entity mention detection via anchor-region networks," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5182–5192. [Online]. Available: https://aclanthology.org/P19-1511

[32] X. Wang et al., "Automated concatenation of embeddings for structured prediction," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 2643–2660, doi: 10.18653/v1/2021.acl-long.206.

[33] S. Bell, H. Yannakoudakis, and M. Rei, "Context is key: Grammatical error detection with contextual word representations," in *Proc. 14th Workshop Innov. Use NLP Building Educ. Appl.*, 2019, pp. 103–115. [Online]. Available: https://aclanthology.org/W19-4410

[34] L. Sun, Y. Sun, F. Ji, and C. Wang, "Joint learning of token context and span feature for span-based nested NER," *IEEE/ACM Trans. Audio, Speech Lang. Proc.*, vol. 28, pp. 2720–2730, 2020, doi: 10.1109/TASLP.2020.3024944.

[35] B. Athiwaratkun, C. N. D. Santos, J. Krone, and B. Xiang, "Augmented natural language for generative sequence labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 375–385. [Online]. Available: https://aclanthology.org/2020.emnlp-main.27

[36] T. Shibuya and E. H. Hovy, "Nested named entity recognition via second-best sequence learning and decoding," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 605–620, 2020, doi: 10.1162/tacl_a_00334.

[37] J. Straková, M. Straka, and J. Hajic, "Neural architectures for nested NER through linearization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5326–5331. [Online]. Available: https://aclanthology.org/P19-1527

[38] Z. Zhao, S. Zhu, and K. Yu, "A hierarchical decoding model for spoken language understanding from unaligned data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7305–7309, doi: 10.1109/ICASSP.2019.8682463.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Comput. Vis. 16th Euro. Conf.*, 2020, pp. 213–229, doi: 10.1007/978-3-030-58452-8_13.

[40] X. Chen, X. Han, J. Hu, F. J. Ruiz, and L. Liu, "Order matters: Probabilistic modeling of node sequence for graph generation," 2021, *arXiv.:2106.06189*.

[41] Z. Tan, Y. Shen, S. Zhang, W. Lu, and Y. Zhuang, "A sequence-to-set network for nested named entity recognition," in *Proc. 13th Int. Joint Conf. Artif. Intell., Virtual Event*, 2021, pp. 3936–3942, doi: 10.24963/ijcai.2021/542.

[42] J. Ye, T. Gui, Y. Luo, Y. Xu, and Q. Zhang, "One2Set: Generating diverse keyphrases as a set," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4598–4608. [Online]. Available: https://aclanthology.org/2021.acl-long.354

[43] P. Yang, F. Luo, S. Ma, J. Lin, and X. Sun, "A deep reinforced sequence-to-set model for multi-label classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5252–5258. [Online]. Available: https://aclanthology.org/P19-1518

[44] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," in *Proc. 4th Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., San Juan, Puerto Rico, May 2-4, 2016. [Online]. Available: http://arxiv.org/abs/1511.06391

[45] K. Qin, C. Li, V. Pavlu, and J. Aslam, "Adapting RNN sequence prediction model to multi-label set prediction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 3181–3190. [Online]. Available: https://aclanthology.org/N19-1321

[46] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[47] H. W. Kuhn et al., "The hungarian method for the assignment problem," in *Proc. 50 Years Integer Program. 1958-2008 - From Early Years State-of-The-Art*, 2010, pp. 29–47, doi: 10.1007/978-3-540-68279-0_2.

[48] S. Pradhan et al., "Towards robust linguistic analysis using ontonotes," in *Proc. 17th Conf. Comput. Natural Lang. Learn.*, 2013, pp. 143–152. [Online]. Available: https://aclanthology.org/W13-3516/

[49] S. Pradhan et al., "Task 1: Share/clef ehealth evaluation lab 2013," in *Proc. Work. Notes CLEF Conf.*, 2013. [Online]. Available: http://ceur-ws.org/Vol-1179/CLEF2013wn-CLEFeHealth-PradhanEt2013.pdf

[50] D. L. Mowery et al., "Task 2: Share/clef ehealth evaluation lab 2014," in *Proc. Work. Notes CLEF Conf.*, 2014, pp. 31–42. [Online]. Available: http://ceur-ws.org/Vol-1180/CLEF2014wn-eHealth-MoweryEt2014.pdf

[51] H. Ji et al., "Overview of TAC-KBP2017 13 languages entity discovery and linking," in *Proc. Text Anal. Conf.*, 2017.

[52] A. Mitchell, S. Strassel, S. Huang, and R. Zakhary, "Ace 2004 multilingual training corpus," 2005. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2005T09

[53] C. Walker, S. Strassel, J. Medero, and K. Maeda, "Ace 2005 multilingual training corpus," 2006. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2006T06

[54] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *Comput. Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[55] E. F. T. K. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Lang.-independent named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn. at HLT-NAACL*, 2003, pp. 142–147. [Online]. Available: https://aclanthology.org/W03-0419/

[56] S. Coope, T. Farghly, D. Gerz, I. Vulic, and M. Henderson, "Span-convert: Few-shot span extraction for dialog with pretrained conversational representations," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 107–121, doi: 10.18653/v1/2020.acl-main.11.

[57] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Proc. Speech Natural Lang.: Proc. Workshop Held Hidden Valley*, 1990, [Online]. Available: https://aclanthology.org/H90-1021/

[58] L. Qin, X. Xu, W. C. He, and T. Liu, "Towards fine-grained transfer: An adaptive graph-interactive framework for joint multiple intent detection and slot filling," in *Proc. Findings Assoc. Comput. Linguistics*, 2020, pp. 1807–1816, doi: 10.18653/v1/2020.findings-emnlp.163.

[59] A. Coucke et al., "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018, *arXiv:1805.10190*.

[60] J. Straková, M. Straka, and J. Hajic, "Neural architectures for nested NER through linearization," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, pp. 5326–5331, doi: 10.18653/v1/p19-1527.

[61] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi, "A general framework for information extraction using dynamic span graphs," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 3036–3046, doi: 10.18653/v1/n19-1308.

[62] J. Wang, L. Shou, K. Chen, and G. Chen, "Pyramid: A layered model for nested named entity recognition," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5918–5928, doi: 10.18653/v1/2020.acl-main.525.

[63] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020, doi: 10.1162/tacl_a_00300.

[64] Z. Jiang, W. Xu, J. Araki, and G. Neubig, "Generalizing natural language analysis through span-relation representations," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 5-10, 2020, pp. 2120–2133, doi: 10.18653/v1/2020.acl-main.192.

[65] G. Paolini et al., "Structured prediction as translation between augmented natural languages," in *Proc. 9th Int. Conf. Learn. Representations Virtual Event*, 2021. [Online]. Available: https://openreview.net/forum?id=US-TP-xnXI

Xiang Chen is currently working toward the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He got the National Scholarship in 2022. He has more than ten publications appeared in several top conferences, such as ICLR, NeurIPS, SIGIR, WWW, IJCAI, EMNLP, and NAACL. His research interests include knowledge graphs and natural language processing. Moreover, he was a PC member of several top conferences and journal, including SIGIR, EMNLP, and TOIS.

Lei Li is currently working toward the M.S. degree with Zhejiang University, Ningbo, China. He got the National Scholarship in 2022. He has authored or coauthored several papers in major international conferences, such as SIGIR, ICLR, and NeurIPS. His research interests include information extraction, multi-modal learning, and language models.

Yuqi Zhu is currently working toward the Ph.D. degree with Zhejiang University, Hangzhou, China. Her research interests include natural language processing, large language model agents, and embodied learning.

Shumin Deng (Member, IEEE) received the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. She is currently a Research Fellow with the National University of Singapore, Singapore. Her research interests include natural language processing, knowledge graph, and knowledge base population in low-resource scenarios. She has obtained 2022 Outstanding Graduate of Zhejiang Province, China, and 2020 Outstanding Intern in Academic Cooperation of Alibaba Group. She was a Reviewer of many prestigious journals, such as IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and *ACM Transactions on Asian Language Information Processing*. She is also a PC Member of many top conferences, such as ICLR, ACL, EMNLP, NAACL, WWW, EACL, AACL, AAAI, and IJCAI.

Chuanqi Tan received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2019. He is currently an Algorithm Expert with Language Technology Lab, Alibaba DAMO Academy. His research interests include question answering, information extraction, and biomedical natural language processing.

Fei Huang is currently a Principal Researcher with Language Technologies Lab, Alibaba DAMO Academy. He leads R&D on NLP foundational technologies, dialogue, and machine translation. His team develops various NLP technologies ranging from lexical, syntactical, semantic, discourse, and deep learning-based algorithms, and integrate them into the Alibaba NLP platform, which supports several hundred internal and external clients with advanced NLP models, systems, and solutions in various industries.

Luo Si is currently a Faculty Member with the Department of Computer Science, Department of Statistics (by courtesy), Purdue University, West Lafayette, IN, USA. He leads a research group working on topics of information retrieval, applied machine learning, and intelligent tutoring. He has authored or coauthored more than 90 journal and conference papers. His research has been supported by the National Science Foundation (NSF), State of Indiana, Purdue University and industry companies such as Yahoo! and Google. He was the recipient of the NSF Career Award in 2008. He is also an Associate Editor for *ACM Transactions on Information System* and *ACM Transactions on Interactive Information Systems*.

Ningyu Zhang (Member, IEEE) is currently an Associate Professor/doctoral supervisor with Zhejiang University, Hangzhou, China, leading the Group about KG and NLP technologies. He has authored or coauthored many papers in top international academic conferences and journals, such as *Nature Machine Intelligence*, *Nature Communications*, NeurIPS, ICLR, AAAI, IJCAI, WWW, KDD, SIGIR, ACL, ENNLP, NAACL, and IEEE/ACM TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE. He was a Senior Program Committee of IJCAI, the Area Chair of ACL, EMNLP, ARR Action Editor, Program Committee of AAAI, NeurIPS, ICLR, WWW, SIGIR, KDD, ICML, AAAI, IJCAI, and an Associate Editor for *Transactions on Asian and Low-Resource Language Information Processing*, and Reviewer of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Knowledge Discovery from Data*, *World Wide Web*, and *Expert Systems with Applications*.

Huajun Chen (Member, IEEE) received the bachelor's and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2000 and 2004, respectively. He is currently a Full Professor with the College of Computer Science and Technologies, Zhejiang University, Hangzhou, China, the Director of Joint Lab on Knowledge Engine, AZFT (Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies), and Deputy Director of the Key Lab of Big Data Intelligence, Zhejiang Province. He was a Visiting Assistant Professor with the Yale Center for Medical Informatics, Yale University, New Haven, CT, USA, from 2006 to 2007, and a Visiting Scholar with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, from 2007 to 2008.