Adversarial Learning for Multi-Task Sequence Labeling With Attention Mechanism

Yu Wang, Yun Li, Ziye Zhu, Hanghang Tong, and Yue Huang

***Abstract*—With the requirements of natural language applications, multi-task sequence labeling methods have some immediate benefits over the single-task sequence labeling methods. Recently, many state-of-the-art multi-task sequence labeling methods were proposed, while still many issues to be resolved including (C1) exploring a more general relationship between tasks, (C2) extracting the task-shared knowledge purely and (C3) merging the task-shared knowledge for each task appropriately. To address the above challenges, we propose MTAA, a symmetric multi-task sequence labeling model, which performs an arbitrary number of tasks simultaneously. Furthermore, MTAA extracts the shared knowledge among tasks by adversarial learning and integrates the proposed multi-representation fusion attention mechanism for merging feature representations. We evaluate MTAA on two widely used data sets: CoNLL2003 and OntoNotes5.0. Experimental results show that our proposed model outperforms the latest methods on the named entity recognition and the syntactic chunking task by a large margin, and achieves state-of-the-art results on the part-of-speech tagging task.**

***Index Terms*—sequence labeling, multi-task learning, adversarial learning, attention mechanism.**

I. Introduction

SEQUENCE Labeling, which aims to label each element of the input sequence with the task label set, is a fundamental task in Natural Language Processing (NLP) [1]. Generally, the sequence labeling methods first encode the text through neural networks [2], and then obtain the label sequence by Conditional Random Fields (CRF) [3] or Softmax decoding [4]. Under these high-level thoughts, single-task sequence labeling with feature embedding and multi-task sequence labeling with knowledge transfer have been the two major encoding objectives [5], [6].

The single-task sequence labeling methods merely treat one task in one training process with the task-specific data set [2], [7]. Normally, the text encoding methods contain various neural networks, including the Convolutional Neural Networks (CNN) [8], Bidirectional Long Short-Term Memory (BiLSTMs) [9] and Transformer [10]. For example, the BiLSTMs-CRF model proposed by Huang et al [11] is capable of completing the Part-of-Speech (POS) tagging, Named Entity Recognition (NER) and Chunking tasks separately. In order to boost the model performance, such methods further utilize additional feature embeddings, and then concatenate them with the original input. For example, Ghaddar and Langlais [12] proposed a variant of the BiLSTMs-CRF model by adding lexical similarity representation, which encodes the similarity of a word to each entity type. Feng et al. [13] combined four additional feature embeddings to cope with the problem of insufficient training data and recognized named entities in low-resource languages. However, these methods cannot guarantee the effectiveness of various additional feature embeddings, because these embeddings bring both the knowledge and noise into the training model. In addition, a single-task sequence labeling model is independently trained for one task at a time, resulting in a significant increase in total training cost for modeling multiple tasks. Unfortunately, downstream NLP tasks, such as knowledge graph construction and question answering system, often require multiple label information in practice. This leads to a gap between the single-task sequence labeling methods and the real-world natural language applications.

In contrast, the multi-task sequence labeling methods learn multiple tasks jointly. As an immediate benefit, this closes the aforementioned gap between real-world applications and single-task sequence labeling methods. Recently, many state-of-the-art multi-task sequence labeling methods have been presented [14], [15], which eliminate the dependency on manually added feature embeddings [16] thanks to transferring shared knowledge among tasks. Some of these methods utilized the source (simple) task to improve the performance of target (challenging) task [17]. For example, Lin et al. [18] exploited the features from high-resource language tagging tasks to improve the low-resource language tagging tasks. However, the dependency on high-quality source tasks and unequal relationship between tasks may limit the generality and applicability of such models. Furthermore, multiple studies [19], [20] have shown that knowledge transfer provides valuable information, but also brings task-specific information (i.e., noise) into the model. In order to filter out noise, several methods [21], [22] employ the adversarial training strategy to learn shared knowledge among tasks. Cao et al. [19] proposed an adversarial transfer learning model for processing NER and Chinese Word Segmentation (CWS) tasks simultaneously.

Manuscript received October 12, 2019; revised March 07, 2020, April 21, 2020 and July 09, 2020; accepted July 15, 2020. Date of publication July 30, 2020; date of current version September 3, 2020. This work was partially supported by Natural Science Foundation of China (No. 61772284), Postgraduate Research & Practice Innovation Program of Jiangsu Province (SJKY19_0766). Hanghang Tong is partially supported by Natural Science Foundation (1947135, 2003924 and 1939725). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfeng Gao. *(Corresponding author: Yun Li.)*

Yu Wang, Yun Li, Ziye Zhu, and Yue Huang are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China (e-mail: 2017070114@njupt.edu.cn; liyun@njupt.edu.cn; 1015041217@njupt.edu.cn; huangyue@njupt.edu.cn).

Hanghang Tong is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL (e-mail: htong@illinois.edu).

Digital Object Identifier 10.1109/TASLP.2020.3013114

In this case, the feature (i.e., knowledge) of word boundary information is partly shared in both tasks. They used a task discriminator to ensure that the shared information extracted from CWS is not mixed with task-specific noise. However, the task discriminator in their model might fail to extract the shared knowledge (referred to as 'discriminator collapse' in this paper) due to the fact that the inputs of the model come from the same language and similar domains (more detail in Section II-B). In addition, current advanced multi-task methods generate multiple feature representations [23], while lacking exploration for appropriately merging these representations. Simply concatenating multiple representations ignores the priority of the representations, which is not suitable for an increasing number of representations.

Based on the above observations, the multi-task sequence labeling method is more suitable for real-world natural language applications [24]. However, it still remains three key challenges have yet to be fully addressed, including (C1) exploring a more general relationship between tasks in end-to-end multi-task methods, (C2) purely extracting task-shared knowledge and (C3) appropriately merging the shared knowledge for each task.

In this paper, we present a multi-task sequence labeling model named MTAA that can effectively address the three challenges mentioned above. We treat the relationship between tasks to be equal to each other. Accordingly, the proposed model is designed as a symmetrical structure to perform an arbitrary number of tasks simultaneously. To purely extract shared knowledge, a variant of the adversarial training strategy is proposed, which effectively alleviates the task discriminator collapse problem. Different from the previous work [19], the goal of the task discriminator in MTAA is to estimate which task-individual encoder the input comes from. To appropriately merge the multiple representations, we present the multi-representation fusion attention mechanism. Specifically, for each sentence, we first encode it into a task-individual representation for each task. Then, the adversarial training modules extract the shared representations among multiple tasks. Furthermore, the multi-representation fusion attention mechanism merges the individual and shared representations with respect to each task. Finally, the task-individual CRFs are employed to decode the label of each sequence labeling task.

The main contributions of our work are as follows,

- The MTAA can perform an arbitrary number of sequence labeling tasks simultaneously, thereby meeting the various needs of different downstream NLP tasks.
- The MTAA extracts shared knowledge among tasks by adversarial learning, while effectively alleviating the task discriminator collapse problem.
- The multi-representation fusion attention mechanism in MTAA model merges the multiple feature representations appropriately.

We evaluate the MTAA model on two public data sets: CoNLL2003 and OntoNotes5.0. Under sets of fair comparison experiments, MTAA outperforms the advanced methods on the NER and Chunking tasks, by a large margin, and achieves state-of-the-art results on the POS tagging task. Especially, MTAA performs more effectively on OntoNotes5.0, which proves that the MTAA has more advantages in dealing with complicated data sets.

II. Problem Statement

In this section, we first define the multi-task sequence labeling problem. Then we analyze the challenges in this task and give some solutions.

A. Problem Definition

Formally, given one or several sentences, the multi-task sequence labeling method learns to generate corresponding label sequences for a group of tasks [18]. We denote T as a set of sequence labeling tasks, and $X = \{x_1, x_2, ..., x_K\}$ as a set of sentences, where K is the number of the sentences in the corpus. For task $t \in T$, $Y^t = \{y_1^t, y_2^t, ..., y_K^t\}$ represents a set of corresponding label sequences. Based on the above notations, we define the multi-task sequence labeling problem as follow,

Problem: Multi-task Sequence Labeling

Given: (1) a set of sentences X, where each sentence $x_i \in X$ contains N_i words. (2) $|T|$ sets of label sequences Y^T, where each set $Y^t \in Y^T$ represents the label information of task t corresponding to the set X. (3) a new sentence $x_{new} \notin X$ which contains N_{new} words;

Find: the label sequences y_{new}^T for the new sentence x_{new} on all tasks T.

Noted that we take the sentence and the corresponding task label sequences as inputs. The goal is to generate the label sequences for a new sentence on all these tasks.

B. Challenges and Ideas

Before we present the details of our MTAA, we first illustrate the three challenges in this task and attach our ideas for each one.

C1: The relationship between sequence labeling tasks. In existing multi-task sequence labeling models, the relationships between tasks are mainly source-target relationship [18], linguistic hierarchy relationship [25], and equal relationship [26]. To be specific, the source-target relationship-based methods achieve significant improvements on the target task by transferring knowledge from source task to target task. The linguistic hierarchy relationship-based methods treat different tasks at different layers where higher layers utilize shared knowledge provided by the lower-level tasks. In contrast, equal relationship-based methods can transfer knowledge in both directions.

Although the methods based on the source-target or linguistic hierarchy relationship perform well in some cases [27], multiple tasks cannot simultaneously make full use of shared knowledge between each other due to the one-way transfer. In addition, the source-target relationship-based methods are based on the hypothesis that the knowledge learned in a related source task can be reused in the target task. Therefore, such methods normally require a high-quality source task to ensure good performance on the target task. Another major limitation

of the linguistic hierarchy relationship-based methods is that we have to subjectively determine the order of tasks. However, it is noted that the order may affect the performance of all tasks.

Idea for C1: Modeling each task equally by the symmetric structure. Several works [26], [28] have reported the shared knowledge between a pair of sequence labeling tasks. Taking the NER, POS tagging and Chunking tasks as an example, the shared knowledge is as follows,

- NER and POS tagging: According to statistics, the entity words recognized in the NER task are generally either nouns or including nouns. Similarly, the noun words recognized in the POS tagging task can be entity words or entity references or entity heads. Obviously, the two tasks are closely related.
- POS tagging and Chunking: For simplicity, the POS tagging task aims to recognize nouns, verbs, adjectives, etc.; the Chunking task aims to recognize noun phrases, verb phrases, etc. It can be seen that the targets of the two tasks are consistent in some respects.
- NER and Chunking: The noun phrases identified in the Chunking task can be considered as coarse-grained entities. Thus, the boundary information extracted in the two tasks is similar, which can effectively enhance the overall performance of both tasks.

Based on these observations, we treat the relationship between tasks to be equal to each other. Accordingly, we design the model to be a symmetric structure that can share knowledge in both directions. This allows the proposed model can perform an arbitrary number of the tasks simultaneously.

C2: Extracting the task-shared knowledge purely. In multi-task learning, models learn various tasks jointly and benefit all of them by transferring knowledge. The knowledge that should be transferred is shared information among tasks, for the reason that the remaining task-specific information might negatively affect the performance of the model [29]. Recently, some multi-task methods [19], [29] adopt an adversarial training strategy (which normally consists of a generator and a discriminator) to purely extract the task-shared knowledge. To be specific, the generator (i.e., task-shared encoder) attempts to extract features shared between a pair of tasks. And the discriminator (i.e., task discriminator) aims to determine which task the extracted features comes from. We consider that the features extracted by task-shared encoder can represent shared knowledge if the task discriminator fails. For example, Cao et al. (2018) [19] proposed an adversarial transfer learning model to process both NER and CWS tasks simultaneously. In specific, they used a task discriminator to ensure that the shared information extracted from CWS without task-specific noise. However, it is difficult for the task discriminator to perform well due to the fact that both inputs of the model belong to the same language and similar domains. This makes the task-shared encoder easy to mislead the task discriminator even if it is not working. As a result, the shared encoder might fail to extract the shared knowledge. This form of failure is called 'discriminator collapse' in this paper.

Idea for C2: A variant of the adversarial training strategy. To purely extract the shared knowledge and alleviate discriminator collapse problem, we adjust the adversarial training strategy in our model. Rather than distinguishing which task corpus the input comes from, we first use the task-individual encoder to encode the input sentence and then distinguish which task-individual encoder the input comes from. The reason for this adjustment is that the task-individual encoder contains more task-related knowledge since it is one of the closest feature extractors to the task decoder (i.e., task-individual CRF). In this way, the task discriminator can successfully alleviate the discriminator collapse problem, thereby ensuring that the task-shared encoder can effectively extract shared knowledge.

C3: Merging the shared knowledge appropriately. The multi-task methods normally extract multiple feature representations for each task including the individual and shared representations [23]. Most of the existing methods simply concatenate these representations or find the mean of all feature representations. As a result, the priority of representations was relatively ignored until recently. Nowadays, appropriately merging multiple representations is becoming increasingly significant for multi-task models.

Idea for C3: Multi-representation fusion attention mechanism. For appropriately merging the shared knowledge, we present a multi-representation fusion attention mechanism, which is a variant of attention mechanism. The traditional attention mechanism [30] consists of a query and a set of key-value pairs, where the query, keys, and values are all vectors. Among them, the query vector is a core component in attention function. In this paper, we use the task-individual representation as to the query in each attention function. The set of key-value pairs is the combination of the individual and shared representations. In this manner, we can integrate the shared knowledge from other tasks with respect to each task.

Combining the symmetric model structure, shared knowledge extraction and fusion methods, our proposed MTAA finally performs equal, pure, and efficient knowledge transfer among tasks, and achieves state-of-the-art performance of multi-task sequence labeling.

III. Model

In this section, we present our multi-task sequence labeling model MTAA, whose overall architecture is depicted in Figure 1. The MTAA contains four main components including task-individual encoder, adversarial training, multi-representation fusion attention mechanism, and task label decoder. For each given sentence, the MTAA first learns the individual representation for each task by the task-individual encoders. Then, the adversarial training modules extract the shared knowledge among tasks, while eliminating the task-specific noise. Subsequently, the multi-representation fusion attention mechanism merges the obtained individual and shared representations with respect to each task. Finally, the task-individual CRF is employed to decode the label of each sequence labeling task. In the following subsections, we describe the main components of our model in detail. To clearly explain the proposed model, we take the most important sequence labeling tasks (i.e., NER, POS tagging, and

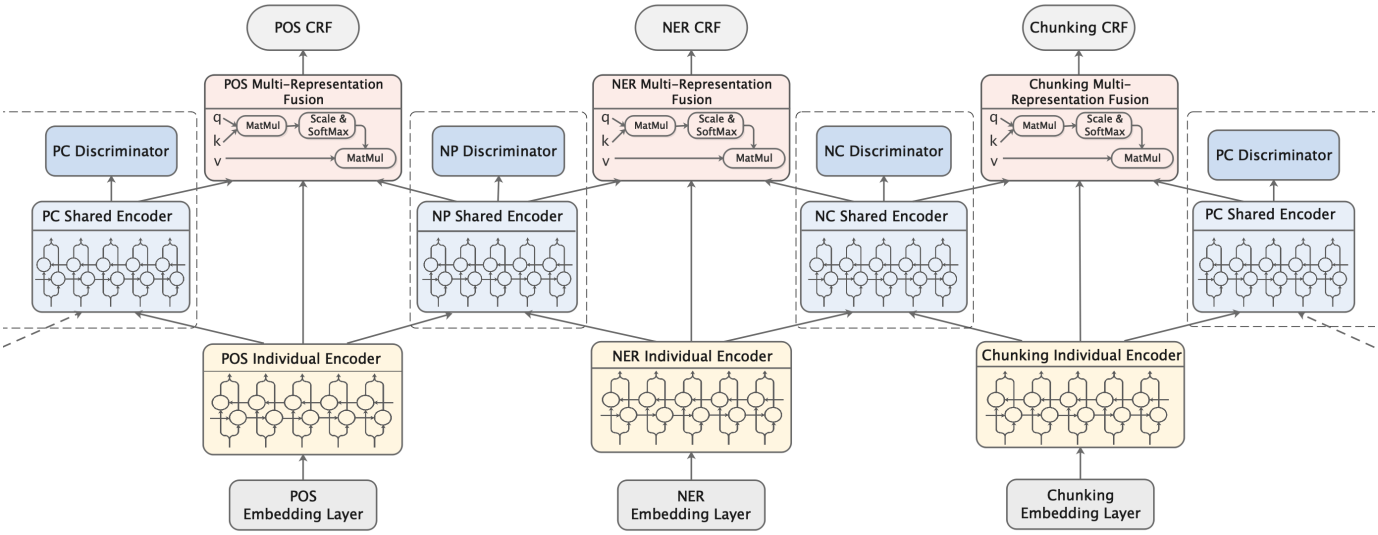


Fig. 1. An overview of our model MTAA. The dotted frame is the adversarial training module. The PC adversarial module on both sides refers to the same.

Chunking tasks) as an example (i.e., $T = \{n, p, c\}$, where n denotes NER, p denotes POS tagging, and c denotes Chunking task).

A. Task-individual Encoder

Given a sentence $x = \{w_1, w_2, ..., w_N\}$, N is the length of the sentence, the task-individual encoder firstly learns the individual features of each task. Specifically, we employ an embedding layer and an encoding layer to encode the sentence into a task-individual representation for each task.

1) Embedding Layer. The embedding layer maps all input words in the sentence into a corresponding embedding sequence. The embedding layer consists of a full pre-trained BERT [10] model and a dense layer. BERT is one of the latest embedding methods, which represents Bidirectional Encoder Representations from Transformers. There are 768 dimensions for each word embedding vector in BERT. In terms of the complexity of the model, we reduced the 768-dimensional embedding to k-dimensional through a simple dense layer. Therefore, as shown in Figure 1, for a given sentence x, the embedding layer converts the sentence x into an individual representation $\mathbf{x}^t = \{\mathbf{w}_1^t, \mathbf{w}_2^t, ..., \mathbf{w}_N^t\}$ for task t, where $\mathbf{w}^t$ is a k-dimensional word vector.

2) Encoding Layer. After embedding the given sentence as a k-dimensional sequence $\mathbf{x}^t$ for task t, we use an encoding layer to learn the task-individual representation for each task. We use the same neural model structure to extract features since all target tasks belong to the sequence labeling. LSTM [31], a particular RNN, is generally adopted to learn the long-term dependencies in many NLP applications. BiLSTMs are able to extract semantic features that reflect the sequential nature of the text. In our model, we built the task-individual encoders with the same BiLSTMs structure but no shared parameters for each task. For simplicity, we denote such a sentence encoding operation as the following equations,

$$\overrightarrow{\mathbf{h}}_i^t = \mathbf{LSTM}_f(\mathbf{w}_i^t, \overrightarrow{\mathbf{h}}_{i-1}^t, \theta_f^t), \tag{1}$$

$$\overleftarrow{\mathbf{h}}_i^t = \mathbf{LSTM}_b(\mathbf{w}_i^t, \overleftarrow{\mathbf{h}}_{i+1}^t, \theta_b^t), \tag{2}$$

$$\mathbf{h}_i^t = \overrightarrow{\mathbf{h}}_i^t \oplus \overleftarrow{\mathbf{h}}_i^t, \tag{3}$$

where $t \in T$, the θ_f^t and θ_b^t denote the parameters of the forward and backward LSTM related to task t, respectively. The $\overrightarrow{\mathbf{h}}_i^t$ and $\overleftarrow{\mathbf{h}}_i^t$ are the k_h-dimensional hidden states at the position i of the forward and backward LSTM, respectively. $\oplus$ denotes concatenation operation. And then we use the $\mathbf{h}^t = \{\mathbf{h}_1^t, \mathbf{h}_2^t, ..., \mathbf{h}_N^t\}$ to denote the individual representation of task t. Through the embedding layer and the BiLSTMs encoding layer, we obtain a task-individual representation for each task.

B. Adversarial Training

In addition to individual representations, we also utilize the adversarial training strategy to purely extract the knowledge shared between a pair of tasks. The adversarial training module normally consists of a generator and a discriminator. The target of the discriminator (i.e., task discriminator) is to determine whether the features extracted by the generator (i.e., task-shared encoder) are shared among tasks.

1) Task-shared Encoder. As we obtain the task-individual representations (i.e., $\mathbf{h}^T = \{\mathbf{h}^n, \mathbf{h}^p, \mathbf{h}^c\}$ in this example) from the task-individual encoders, a task-shared encoder is used to learn the shared representation between a pair of individual representations. Similar to the task-individual encoder, we also utilize the BiLSTMs structure as the task-shared encoder. Different from previous works [19] use representations from embedding layers as the inputs, the benefit of using representations from task-individual encoders is to alleviate the discriminator collapse problem. For simplicity, we denote the BiLSTMs as **S**, and the task-shared feature extraction follows the equation,

$$\mathbf{s}_{\alpha}^{t} = \mathbf{S}(\mathbf{h}^{t}, \theta_{s}^{\alpha}), t \in \alpha \tag{4}$$

where θ_s^α denotes the parameters of the task-shared encoder set for the pair of tasks α (where $\alpha \in \{\{n,p\},\{n,c\},\{p,c\}\}$ in this example), and $\mathbf{s}_\alpha^t$ denotes shared representation for task t extracted by the shared encoder with parameters θ_s^α. Taking the encoder shared by NER and POS as an example, we use the $\mathbf{s}_{\{n,p\}}^n$ to represent NER shared representation and $\mathbf{s}_{\{n,p\}}^p$ to represent POS shared representation. It should be mentioned that a shared encoder alternately receives the input from two related tasks (i.e., one task-individual representation at each time).

2) Task Discriminator. In our model, task discriminators also share the same neural network structure but do not share parameters. The task discriminator uses the task-shared representation to estimate which task-individual encoder the input comes from. Specifically, we adopt a two-layer multilayer perceptron networks (MLP) as the task discriminator to achieve the probability distributions over each sequence. Formally, the equations can be expressed as follow,

$$\mathbf{D}(\mathbf{s}_{\alpha}^{t}, \theta_{d}^{\alpha}) = \mathrm{softmax}(\mathrm{MLP}(\mathbf{s}_{\alpha}^{t})), \tag{5}$$

where the α denotes a pair of tasks (where $\alpha \in \{\{n,p\},\{n,c\},\{p,c\}\}$ in this example). For simplicity, θ_d^α denotes the parameters of each task discriminator. Contrary to the task discriminator, the task-shared encoder is expected to generate the representation that can mislead the task discriminator. Therefore, the adversarial training process is a min-max game and can be formalized as follow,

$$\mathcal{L}_{adv} = \sum_{\alpha} \min_{\theta_s^\alpha} (\max_{\theta_d^\alpha} \sum_{t}^{\alpha} \sum_{k}^{K} \log \mathbf{D}(\mathbf{S}(\mathbf{h}_{(k)}^{t}))), \tag{6}$$

where the θ_s^α and θ_d^α denote the trainable parameters of task-shared encoder and task discriminator for task pair α. The K is the number of training examples and $\mathbf{h}_{(k)}^t$ is the k-th example of task t.

It should be noted that the task-shared encoder (i.e., generator) can extract the knowledge between a pair of tasks. And the task discriminator is used to eliminate the task-specific noise. After sufficient training, the task-shared encoder and the task discriminator reach a balance. Thereby, the representation generated by task-shared encoder represents the shared features between a pair of tasks in which task-specific noise has been filtered.

C. Multi-Representation Fusion Attention Mechanism

At this position, we have obtained an individual representation and multiple shared representations for each task. We propose the multi-representation fusion attention mechanism to appropriately merge these representations. The proposed multi-representation fusion attention mechanism consists of a query and a set of key-value pairs, where the query, keys and values are all vectors. The query vector is the core component in the attention function [30]. In our MTAA model, the task-individual representation of each task is the most basic of these representations. Hence, we use individual representation as the query in the attention function. For task t, the set of key-value $\mathbf{e}^t$ is the combination of the individual representation $\mathbf{h}^t$ and shared representations $\mathbf{s}^t$. For example, the key-value set of NER task should be $\mathbf{e}^n = \{\mathbf{h}^n, \mathbf{s}_{\{n,p\}}^n, \mathbf{s}_{\{n,c\}}^n\}$ (pair of task $\{p,c\}$ is not related to NER task). The attention score is defined as follow,

$$\beta_i^t = \frac{\exp(\mathbf{h}^t, \mathbf{e}_i^t)}{\sum_{\mathbf{e}_j^t \in \mathbf{e}^t} \exp(\mathbf{h}^t, \mathbf{e}_j^t)}, \mathbf{e}_i^t \in \mathbf{e}^t. \tag{7}$$

The attention scores are referred to estimate how well those representations related to the task-individual representation. It can be used to compute the fusion representation $\bar{\mathbf{e}}^t$ as

$$\bar{\mathbf{e}}^t = \sum_{\mathbf{e}_i^t \in \mathbf{e}^t} \beta_i^t \mathbf{e}_i^t. \tag{8}$$

It should be mentioned that part of the features among all representations is overlapping. For instance, the word boundary feature is not only an individual feature but also a shared feature. Moreover, The three representations might all contain the word boundary feature. The repetitive features cannot improve the performance of our model, and make the attention mechanism invalid. Therefore, we adopt orthogonality constraints [32] to address the *overlapping issue* before the attention mechanism. We minimize the penalty function as follow,

$$\mathcal{L}_o = \sum_{t}^{T} \| \sum_{\alpha} (\boldsymbol{I}^t)^T \boldsymbol{S}_\alpha^t \|_F, \tag{9}$$

where the $\boldsymbol{I}^t$ is the matrix whose row vectors are the individual representation task t, the $\boldsymbol{S}_\alpha^t$ is the matrix whose row vectors are the related shared representations of task t (in this example, $\alpha \in \{\{n,p\},\{n,c\}\}$ is related to NER task, and pair $\{p,c\}$ is not related to NER task). And the $\|\cdot\|_F$ is the squared Frobenius norm [33].

Noted that, we eventually construct the fusion representation $\bar{\mathbf{e}}^t$ for each sequence labeling task. These fusion representations will be the input to the task label decoder.

D. Task Label Decoder

Finally, we employ a task-individual CRF to decode the representation for each sequence labeling task. For task t, the CRF use the $\bar{\mathbf{e}}^t$ to predict the task label sequence. The scoring equation defined by CRF is calculated as follow,

$$\begin{aligned} score^t(x, y^t) = \sum_{i=1}^{N} (&\log \psi_{EMIT}^t(y_i^t \to w_i) \\ &+ \log \psi_{TRANS}^t(y_{i-1}^t \to y_i^t)), \end{aligned} \tag{10}$$

where the $\psi_{EMIT}^t(y_i^t \to w_i)$ comes from the $\bar{\mathbf{e}}_i^t$ at timestep i. It represents the emission potential from the word w_i to the tag y_i^t. The $\psi_{TRANS}^t \in \mathbb{R}^M$ is the transition matrix, which comes from CRF, and M is the tag size. Moreover, $\psi_{TRANS}^t(y_{i-1}^t \to y_i^t)$ controls the transition probability from y_{i-1}^t to y_i^t. Therefore, we can optimize the sequence label tasks with the following equation,

$$\begin{aligned} \mathcal{L}_t &= -\log(p(y^t|x)) \\ &= -\log(\frac{\exp(score^t(x, y^t))}{\sum_{y'^t \in Y_x^t} \exp(score^t(x, y'^t))}), t \in T. \end{aligned} \tag{11}$$

Algorithm 1: Training process of the MTAA model

Input: Training data ($\chi = \{X, Y^T\}$), where Y^T is the ground truth label sequences; learning rate(η).
Initialization: Initialize parameters (θ^n, θ^p, θ^c, θ_s^{np}, θ_s^{nc}, θ_s^{pc}, θ_d^{np}, θ_d^{nc}, θ_d^{pc}, $\mathbf{A}^n$, $\mathbf{A}^p$, $\mathbf{A}^c$, $\mathbf{M}^n$, $\mathbf{M}^p$, $\mathbf{M}^c$).
Output: (θ^-, $\mathbf{A}^t$, $\mathbf{M}^t$).
for each $(x, y^T) \in \chi$ **do**
  generate three individual representations $\{\mathbf{h}^n, \mathbf{h}^p, \mathbf{h}^c\}$ for x;
  generate two shared representations $\{\mathbf{s}^n_{\{n,p\}}, \mathbf{s}^n_{\{n,c\}}\}$ and predict their task label for $\mathbf{h}^n$;
  generate two shared representations $\{\mathbf{s}^p_{\{n,p\}}, \mathbf{s}^p_{\{p,c\}}\}$ and predict their task label for $\mathbf{h}^p$;
  generate two shared representations $\{\mathbf{s}^c_{\{n,c\}}, \mathbf{s}^c_{\{p,c\}}\}$ and predict their task label for $\mathbf{h}^c$;
  transform all representations into the $\bar{\mathbf{e}}^n$, $\bar{\mathbf{e}}^p$ and $\bar{\mathbf{e}}^c$;
  compute the label sequences y^n, y^p and y^c;
  update parameters (θ^n, θ^p, θ^c, θ_s^{np}, θ_s^{nc}, θ_s^{pc}, θ_d^{np}, θ_d^{nc}, θ_d^{pc}, $\mathbf{A}^n$, $\mathbf{A}^p$, $\mathbf{A}^c$, $\mathbf{M}^n$, $\mathbf{M}^p$, $\mathbf{M}^c$);
end for
return (θ^-, $\mathbf{A}^t$, $\mathbf{M}^t$)

E. Training

The final objective function of our proposed model is defined as follow,

$$\mathcal{L} = \sum_{t}^{T} \mathcal{L}_t + \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_o, \tag{12}$$

where T denotes a set of sequence labeling tasks (i.e., NER, POS tagging and Chunking tasks in this example), λ_1 and λ_2 are the hyper-parameters. In this case, we use λ_1 to increase the proportion of the adversarial training loss in the total loss, because the CRFs' loss is much larger than the adversarial training loss in our experiments. We choose the Adaptive Moment Estimation (Adam) [34] method to optimize the MTAA. It should be noted that, regardless of the model structure or training strategy, the MTAA will treat each sequence labeling task equally.

The full optimization procedure for the MTAA is shown in Algorithm 1, where θ^-, $\mathbf{A}^t$ and $\mathbf{M}^t$ denote the parameters of neural networks, attention matrices, and CRFs, respectively.

IV. Experiments

In this section, we first introduce the data sets used for evaluation and then present the experimental setup and baseline methods. Finally, we present the main experimental results and some detailed analysis of our MTAA model.

A. Data Sets

To evaluate the performance of our proposed MTAA model, we perform experiments on two widely used data sets, including CoNLL2003 [35] and OntoNotes5.0 [36]. Table I shows the size of sentences, tokens, and labels for training, validation, and test sets for each data set.

The CoNLL2003 is a classic data set in the NLP field, which has a training file, a valid file, and a test file. In our experiments, we only use the English data of CoNLL2003. It was taken from the Reuters Corpus which includes 1,393 English Reuters news reports between August 1996 and August 1997. For all data, a tokenizer, a POS tagger, a chunker and named entity tag were applied to the raw data. The English data was tagged and chunked by the memory-based MBT tagger [37]. It contains ten types of chunking and over thirty types POS tags. Named entity tagging was done with four entity types by hand at the University of Antwerp.

With the development of natural language applications, there is an urgent need for richer semantic representations. To address this challenge, the OntoNotes5.0 was provided by the OntoNotes project [38], which is a corpus of large-scale, accurate, and integrated annotation of multiple levels of the shallow semantic structure in text. According to the type of linguistic annotation that represents, OntoNotes5.0 is divided into six logical blocks: (i) Treebank, (ii) PropBank, (iii) Word Sense, (iv) Names, (v) Coreference and (vi) Ontology. The data set comprises various genres (newswire, magazine articles, broadcast news, broadcast conversations, web data, and conversational speech) in three languages (English, Chinese, and Arabic). Specifically, it includes roughly 1.5 million words of English, 800K of Chinese, and 300K of Arabic. In our experiments, we focus on the English and Chinese data of OntoNotes5.0. Besides, according to the tag format of CoNLL2003, we process the tags of OntoNotes5.0 and obtain the POS, chunking and named entity tags for all English and Chinese data of OntoNotes5.0.

B. Experimental Setup

In our experiments, we use 128-dimensional word embeddings. The word embeddings are obtained by the full pre-trained BERT followed by a dense layer. All trainable parameters in our model are initialized by the method described by Glorot and Bengio [39]. We train our model by the Adam optimizer [34] with gradient clipping of 5 [40], and implement it under PyTorch [1].

In addition, we assign the hyper-parameters, which are reported in Table II, using the default values based on the experience. We construct each individual encoder and shared encoder by two BiLSTMs layers with 256 units for considering the trade-off between implementation complexity and model performance. The initial learning rate α is set to 0.008 and decreases as the training steps increase. The batch size is set to 64 at the sentence level. To avoid model over-fitting problem, we apply Dropout [41] to the output of LSTM layer at a rate of 0.5. We monitor the training process on the validation set and report the final result on the test set. All of our experiments are performed on NVIDIA 1080ti GPU and Intel i7-8700K CPU. The training time of MTAA on the CoNLL2003 data set is 50 minutes each epoch, and the decoding (i.e., testing) time is 4 minutes. From the performance on the validation set, our model reached the best performance after 20 epochs.

[1]https://pytorch.org/

TABLE I
THE STATISTICS OF THE DATA SETS.

	Training		Validation		Test	
	sentence #	token #	sentence #	token #	sentence #	token #
CoNLL2003	14,987	204,567	3,644	51,578	3,486	46,666
OntoNotes5.0 (English)	81,828	1,088,503	11,066	157,724	11,257	172,728
OntoNotes5.0 (Chinese)	37,746	751,902	5,586	111,756	4,472	92,142

TABLE II
HYPER-PARAMETERS USED FOR TRAINING THE MTAA MODEL.

Layers	Hyper-parameters	
BERT	dimension	768
Dense	hidden size	128
Individual Encoder	hidden size & layer	256 & 2
Shared Encoder	hidden size & layer	256 & 2
Discriminator	hidden size & layer	256 & 2
Attention layer	hidden size	256
Dropout	rate	0.5
λ_1	rate	5
λ_2	rate	0.02

C. Baseline Methods

As our model can accomplish multiple sequence labeling tasks in one training process, we compare it with the latest models of NER, POS tagging, and Chunking tasks on both data sets. We first introduce the following single-task sequence labeling methods, which can only accomplish on specific tasks:

- LSTM-CRF [2]: a well-known neural model for NER task, which uses the BiLSTMs with a sequential CRF layer above it.
- LSTM-CNNs-CRF [5]: an end-to-end sequence labeling system combining the BiLSTMs, CNN and CRF, without the need for feature engineering or data pre-processing.
- NCRF++: [42]: a mature PyTorch-based toolkit for general sequence labeling tasks. It uses CNN and BiLSTMs to learn both character and word sequence representations for sequence labeling.
- TagLM [43]: a language-model-augmented sequence tagger, which utilizes additional pre-trained context embeddings from bidirectional language models.
- CSE [44]: a novel type of word embedding, short for Contextual String Embeddings, which is produced by a trained character language model.
- ELMo [45]: a type of deep contextualized word representation, short for Embeddings from Language Model, which provides multi-sense information for downstream NLP tasks in addition to word syntax and semantics.
- BERT [10]: a new language representation model, short for Bidirectional Encoder Representations from Transformers, which extends transfer learning with language models from deep unidirectional architectures to deep bidirectional architectures.
- Joint-Yang [46]: a sequence labeling method, which utilizes the discrete manual feature to complement the features automatically induced from neural networks.
- Lattice LSTM [47]: a lattice-structured LSTM model for Chinese NER, which explicitly leverages word and word sequence information to avoid segmentation errors.

To further explore the effectiveness of MTAA as a multi-task model, we compare it with the following state-of-the-art multi-task sequence labeling methods:

- Transfer model [16]: a neural sequence taggers method based on transfer learning, which can be used for tasks without insufficient training data.
- JMT [25]: a joint many-task model, which leverages linguistic levels of morphology, syntax and semantics to solve increasingly complex tasks.
- CVT [48]: a self-training algorithm suitable for neural sequence model, short for Cross-View Training, which leverages both labeled and unlabeled data to improve the representations of sentence encoder.
- SC-LSTM [28]: a new LSTM cell, short for Shared-Cell LSTM, which can simultaneously learn task-shared and specific information.

D. Main Results

The main experimental results of our proposed MTAA in NER, Chunking and POS tagging tasks on the CoNLL2003, OntoNotes5.0 (English) and OntoNotes5.0 (Chinese) are illustrated in Table III, Table IV and Table V, respectively. Since the embedding methods used by these compared multi-task sequence labeling methods are varied, we also provide the experimental results of MTAA with corresponding embedding methods for a fair comparison. It should be mentioned that most of these methods (e.g., ELMo [45] and CSE [44]) cannot be re-implemented in Chinese since the Chinese characters are different from characters in English or pre-trained model does not support Chinese. Therefore, we only provide comparison results on CoNLL2003 and OntoNotes5.0 (English) data sets, shown in Table III and Table IV. In order to further demonstrate the effectiveness of our proposed MTAA in other languages, we compare our model with several state-of-the-art methods on OntoNotes5.0 (Chinese), and the corresponding experimental results are listed in Table V. Since there are few methods for Chinese Chunking and POS tagging tasks, we only report results on the NER task in this set of experiments.

TABLE III
COMPARISON OF THE LATEST METHODS FOR NER, CHUNKING, AND POS TAGGING TASKS IN F1 SCORE ON CONLL2003. * DENOTES RANDOMLY INITIALIZED EMBEDDING METHOD, ○ DENOTES ELMO EMBEDDING METHOD, AND † DENOTES MULTIPLE AUXILIARY PREDICTION MODULES.

Model	NER	Chunking	POS
LSTM-CRF [2]	90.94	94.62	95.75
LSTM-CNNs-CRF [5]	91.21	94.79	95.82
NCRF++ [42]	91.35	95.03	97.08
TagLM [43]	91.93	95.32	-
CSE [44]	93.09	96.04	**97.42**
ELMo [45]	92.22	95.39	96.43
BERT [10]	92.40	95.48	96.57
JMT* [25]	-	95.34	97.24
Transfer Model* [16]	91.26	95.11	97.19
CVT*† [48]	92.61	96.15	97.34
SC-LSTM○ [28]	92.6	96.27	96.64
$\text{MTAA}^{*}_{\text{Basic}}$	91.92	95.48	96.84
$\text{MTAA}^{\circ}_{\text{ELMo}}$	92.60	96.67	97.11
MTAA	**93.45**	**96.91**	97.28

TABLE IV
COMPARISON OF LATEST METHODS FOR NER, CHUNKING, AND POS TAGGING TASKS IN F1 SCORE ON ONTONOTES5.0. * DENOTES RANDOMLY INITIALIZED EMBEDDING METHOD, ○ DENOTES ELMO EMBEDDING METHOD AND † DENOTES MULTIPLE AUXILIARY PREDICTION MODULES.

Model	NER	Chunking	POS
CSE [44]	89.71	87.93	**96.72**
ELMo [45]	86.72	86.85	96.13
BERT [10]	87.95	87.77	96.29
CVT*† [48]	88.81	87.96	96.32
SC-LSTM○ [28]	87.66	88.02	96.12
$\text{MTAA}^{*}_{\text{Basic}}$	88.26	87.26	95.97
$\text{MTAA}^{\circ}_{\text{ELMo}}$	88.94	88.13	96.25
MTAA	**89.83**	**88.44**	96.38

TABLE V
COMPARISON OF LATEST METHODS FOR NER IN F1 SCORE ON ONTONOTES5.0 (CHINESE).

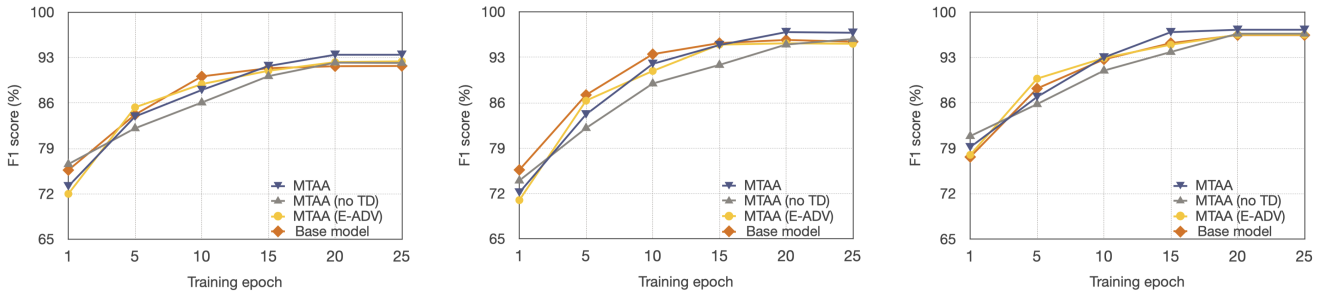
Model	NER
Lattice [47]	75.72
Joint-Yang [46]	76.34
MTAA	**76.86**

From those tables, we can first find that MTAA achieves distinct improvements over all comparison methods in the NER and Chunking tasks, and obtains competitive results in POS tagging task. We further observe that the results obtained by all methods on the OntoNotes 5.0 (English) data set are slightly lower results than those on CoNLL2003. The reason might be that OntoNotes5.0 is a more complex corpus and requires more sequential features. To analyze the experimental results, the subsequent content is arranged according to each task.

Although the NER task is one of the most challenging tasks in sequence labeling problem, MTAA outperforms all compared single-task models on both data sets, which indicates the effectiveness of exploiting shared knowledge among tasks. For example, MTAA surpasses ELMo [45] by 1.23%, BERT [10] by 1.05%, CSE [44] by 0.36% on the CoNLL2003, and also yields 3.11%, 1.88% and 0.12% improvements on the OntoNotes5.0. All of the three methods train each task separately and leverage contextual embeddings. For example, CSE [44] generates the contextual string embeddings by training character-level language model. The main limitation of these methods is the time cost of training the language model. Our proposed MTAA can achieve significant improvements and flexibly utilize pre-trained models. Therefore, this demonstrates the effectiveness of the knowledge transfer and multi-representation fusion attention mechanism designed in our model. In addition, compared with the multi-task models, MTAA is still significantly superior to others. Although CVT [48] (i.e., the closest competitor) contains multiple auxiliary prediction modules, MTAA still surpasses it by 0.84% on CoNLL2003 and 1.02% on OntoNotes. Similarly, the $\text{MTAA}_{\text{ELMo}}$ also surpasses SC-LSTM [28] in NER task on both data sets. It should be noted that the structures of CVT and SC-LSTM are similar to MTAA, but neither of them considers eliminating task-specific noise. These results once again support that MTAA is more effective than others in purely extracting shared knowledge. Furthermore, we find from Table V that our MTAA outperforms other approaches tested in the Chinese NER task. These results support that our MTAA can perform well in other languages.

For the Chunking task, most advanced methods obtain F1 scores higher than 95% on CoNLL2003 data set. These results indicate that Chunking task is less challenging than NER task. We can find that $\text{MTAA}_{\text{ELMo}}$ outperforms its best competitor (i.e., SC-LSTM) in F1 score by 0.40% on CoNLL2003 and 0.11% on OntoNotes. In addition, MTAA surpasses all the comparison methods and achieves 96.91% and 88.84% in F1 score on the two data sets. These significant improvements show that shared knowledge transferred from other tasks is most helpful to the Chunking task. One finding worth considering is that most of comparison methods cannot perform well on the Chunking task of OntoNotes5.0. The main reason is that the labels of Chunking task on OntoNotes5.0 are extremely complex, which include tree structure information.

Focus on the results on the POS tagging task, we can observe that the accuracy of the most state-of-the-art methods are close to 97%. It is very common for these POS tagging methods to achieve such results on certain specific data set (e.g., Penn Treebank). However, the proposed MTAA still significantly outperforms most methods on both data sets. For example, MTAA defeats CTV [48], the closest competitor in the multi-task sequence labeling methods, with 0.06% improvement on the OntoNotes data set. Meanwhile, our model achieves the second-best performance compared with all single-task sequence labeling methods on the OntoNotes5.0, of which only CSE [44] outperforms MTAA. We consider



(a) NER (b) Chunking (c) POS

Fig. 2. The effectiveness of the adversarial training strategy. Base Model denotes the single-task version of MTAA. MTAA(no TD) denotes MTAA without task discriminator. MTAA(E-ADV) denotes that the adversarial training module is placed after the embedding layer.

TABLE VI
EFFECT OF ADVERSARIAL TRAINING STRATEGY (F1 SCORE).

Model	NER	Chunking	POS
Base Model	91.72	95.43	96.45
MTAA(no TD)	92.18	95.84	96.67
MTAA(E-ADV)	92.43	95.11	96.51
MTAA	**93.45**	**96.91**	**97.28**

that the CSE [44] achieves such great success by leveraging character-level language model. We will investigate whether MTAA can leverage the contextual string embeddings in future work.

Overall, our proposed MTAA achieves state-of-the-art performances on all three sequence labeling tasks. It should be noted that these results are obtained without any hand-crafted features such as capitalization, prefixes, and suffixes. In addition, the performance of MTAA on OntoNotes5.0 is better than that on CoNLL2003. In terms of the length of sentences and the number of complex sentences, OntoNotes5.0 is more complicated than CoNLL2003. Therefore, these results indicate that our model has more advantages in dealing with complicated data set.

E. Ablation Study

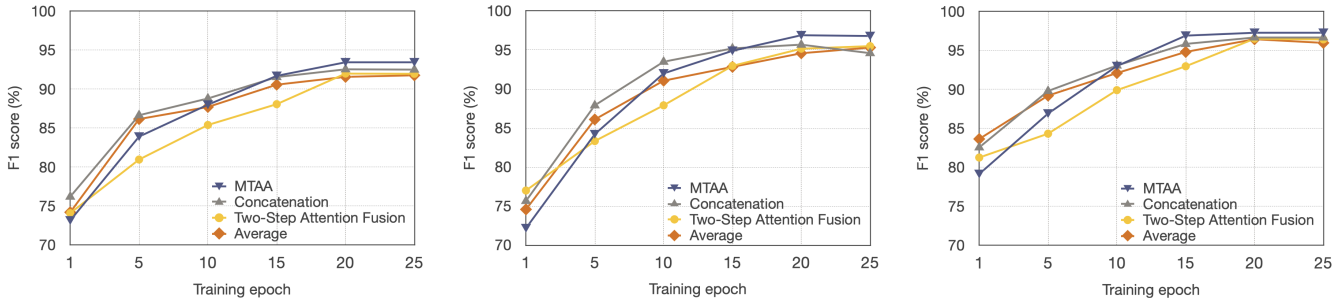
In this section, we conduct a series of ablation experiments on the CoNLL2003 to quantify the contributions of the proposed modules in MTAA.

1) Effect of Adversarial Training Strategy. In our model, we exploit an adversarial training strategy to extract task-shared knowledge while reducing the task-specific noise. To investigate the effect of shared knowledge extraction, we compare the performance of different extraction methods in this set of experiments, and the results are shown in Figure 2 and Table VI. Especially, the base model in Table VI denotes the single-task version of our MTAA, which removes the modules related to knowledge transfer. The MTAA(no TD) denotes MTAA without task discriminator, that is, task-specific noise is not filtered during the knowledge transfer process. MTAA(E-ADV) denotes that the adversarial training module in MTAA is placed after the embedding layer, which is designed to explore the effect of the adversarial training strategy in different locations.

From the Figure 2, we can first observe that shared knowledge has a great impact on the performance of all tasks. In almost all tasks, the other three transfer methods are generally superior to the base model. Moreover, MTAA(E-ADV) achieves better performance than MTAA(no TD) and MTAA in the first five epochs of training. However, the MTAA(no TD) can surpass MTAA(E-ADV) on both Chunking and POS tagging tasks at the end of training. The results turn out that MTAA(E-ADV) might suffer from the task discriminator collapse problem, while the task-shared encoder can easily mislead the task discriminator. The main reason we consider is that these three tasks belong to sequence labeling tasks, and their inputs come from the same sentence. Therefore, it is hard to present the respective characteristics of different tasks only after the word embedding.

As shown in Table VI, compared with the base model, MTAA(no TD) achieves 0.46%, 0.41% and 0.22% improvements on NER, Chunking and POS tagging tasks respectively. Further investigation finds that MTAA improves MTAA(no TD) by 0.61-1.27% on all tasks. The results indicate that with the task discriminator is removed, the task-shared encoder might bring task-specific noise into the model, resulting in a negative transfer effect. Compare with MTAA(E-ADV), our proposed MTAA significantly achieves 0.77-1.80% improvements on all tasks. Thus, the adversarial training strategy can effectively improve the performance of the model for multi-task learning, and the location of the adversarial training is also important.

2) Effect of Fusion Methods. In our model, we present the multi-representation fusion attention mechanism to dynamically merge individual and shared representations. To study the effect of our proposed fusion method, we use other fusion methods to replace it in this set of experiments, and the results are shown in Figure 3 and Table VII. Inspired by exploiting the feature embedding methods, we concatenate and average the obtained representations, respectively. Furthermore, we design a two-step attention mechanism in the experiments. The two-step attention mechanism means that the individual representation can be merged with one of the shared representations into the middle representation by the traditional



(a) NER (b) Chunking (c) POS

Fig. 3. The effectiveness of multi-representation fusion attention mechanism. Average denotes that we take the average of the individual and shared representations instead of multi-representation fusion attention mechanism. Concatenation denotes that we concatenated the individual and shared representations instead of multi-representation fusion attention mechanism. Two-step Attention means another proposed two-step attention mechanism.

TABLE VII
EFFECT OF FUSION METHODS (F1 SCORE).

Model	NER	Chunking	POS
Average	92.78	94.11	95.98
Concatenation	93.01	95.33	96.67
Two-Step Attention Fusion	92.97	95.51	96.42
MTAA	**93.45**	**96.91**	**97.28**

TABLE VIII
COMPARISON OF DIFFERENT EMBEDDING METHODS USED IN BASE MODEL AND MTAA (F1 SCORE).

Model	Embedding	NER	Chunking	POS
Base Model	Basic	90.18	93.17	95.68
	ELMo	91.36	95.27	96.12
	BERT	91.72	95.43	96.45
MTAA	Basic	91.92	95.48	96.84
	ELMo	92.60	96.67	97.11
	BERT	**93.45**	**96.91**	**97.28**

attention mechanism each time. After that, we can get multiple middle representations. Then, we merge them into the fusion representation by the traditional attention mechanism.

From Figure 3, one could observe that both concatenation and average methods are generally superior to the attention-based methods in the first 10 epochs on all tasks. The main reason is that the static fusion method (i.e., concatenate and average methods) without additional parameters obtain better training efficiency. Although the two-step attention mechanism contains the most parameters in these comparison methods, our multi-representation fusion attention mechanism still achieves better performance than it. So the essential reason is that the two-step calculation is too complicated which leads to ineffective fusion. From the results in Table VII, we can see that the concatenation method is better than the average method on all tasks, and fusion attention mechanism we proposed is more effective than both of them. The results demonstrate that static fusion methods, which simply merge the multiple representations independently, cannot perform well in the multi-task learning model. The results again support the effectiveness of our proposed multi-representation fusion method in multi-task learning.

F. Analysis of Embedding Methods

In our MTAA, the embedding layer consists of a full pre-trained BERT model and a dense layer. To explore the effectiveness of the MTAA with different embedding methods, we examine the MTAA and base model (same as defined in Ablation Study) with different embedding methods, including basic embedding (i.e., randomly initialized word embeddings and character embeddings), pre-trained ELMo and pre-trained BERT. This set of experiments is performed on CoNLL2003, and the results are shown in Table VIII. From the results, we observe that the models with pre-trained ELMo always outperform the models with only basic embedding. For example, the base model with ELMo improves basic embedding by 0.54-2.10% on all tasks, and the performance improvements on MTAA with EMLo are also significant. Moreover, MTAA with EMLo surpasses the base model with BERT by 0.66-1.24% on all tasks.

In addition, the results in Table VIII show that the BERT achieves the best performance of both base model and MTAA. To be specific, for the base model, the BERT outperforms the basic embedding by 0.77-2.26% on all tasks. And it also achieves significant improvements for MTAA, such as MTAA with BERT surpasses MTAA with basic embedding by 0.44-1.53% on all tasks. From Table VIII, one could also observe that with the help of embedding methods (i.e, ELMo and BERT), both the base model and MTAA can be significantly improved. This observation supports that the knowledge from the external corpus can improve the performance of the model on the target task. Moreover, the improvement of using the pre-trained embedding methods on MTAA is slightly less than that of the base model, because the MTAA can transfer shared knowledge by itself.

G. Case Study

To further illustrate the effectiveness of our proposed model in purely extracting shared knowledge, we take a sentence

TABLE IX
AN EXAMPLE OF PREDICTED RESULTS IN CONLL2003 TEST DATA SET. THE P, C, AND N DENOTE THE POS TAGGING, CHUNKING, AND NER TASKS, RESPECTIVELY. THE INCORRECT LABELS ARE HIGHLIGHTED IN RED.

Model	**Task**	West	Indies	batsman	Brian	Lara	suffered	another	blow	to	his	Australian	tour
Golden	P	NNP	NNP	NN	NNP	NNP	VBD	DT	NN	TO	PRP	JJ	NN
	C	B-NP	I-NP	I-NP	I-NP	I-NP	B-VP	B-NP	I-NP	B-PP	B-NP	I-NP	I-NP
	N	B-LOC	I-LOC	O	B-PER	I-PER	O	O	O	O	O	B-MISC	O
Base Model	P	NNP	NNP	NNP	NNP	NNP	VBD	DT	NNP	TO	PRP	JJ	NN
	C	B-NP	I-NP	I-NP	I-NP	I-NP	B-VP	B-NP	I-NP	B-PP	B-NP	I-NP	I-NP
	N	B-LOC	I-LOC	O	B-PER	I-PER	O	O	O	O	B-MISC	I-MISC	I-MISC
MTAA(no TD)	P	NNP	NNP	NNP	NNP	NNP	VBD	DT	NN	TO	PRP	JJ	NN
	C	B-NP	I-NP	I-NP	B-NP	I-NP	B-VP	B-NP	I-NP	B-PP	B-NP	I-NP	I-NP
	N	B-LOC	I-LOC	I-LOC	B-PER	I-PER	O	O	O	O	O	B-MISC	O
MTAA	P	NNP	NNP	NN	NNP	NNP	VBD	DT	NN	TO	PRP	JJ	NN
	C	B-NP	I-NP	I-NP	I-NP	I-NP	B-VP	B-NP	I-NP	B-PP	B-NP	I-NP	I-NP
	N	B-LOC	I-LOC	O	B-PER	I-PER	O	O	O	O	O	B-MISC	O

in CoNLL2003 test set as an example, as shown in Table IX. From this table, we can see that 'West Indies batsman Brian Lara' contains a location entity and a person entity, and these five words form an NP-phrase. For these five words, the MTAA(no TD) fails on all three labeling tasks and the base model fails on the POS task. These results demonstrate that the full knowledge transfer manner cannot achieve good performance due to the task-specific noise. This example again supports the effectiveness of adversarial training in filtering task-specific noise.

More importantly, this sentence is also a positive example of presenting the shared knowledge among tasks. Such as, the labeling results of 'West Indies' reflect the consistency among the nouns of the POS tagging task, the noun phrase of the Chunking task, and the person entity of the NER task. Moreover, one could observe that the targets of the POS tagging and Chunking tasks are consistent in some respects from 'suffered another blow' (e.g., the tag 'VBD' and 'B-VP'). These results indicate that the relationship between tasks should be equal in order to transfer knowledge in both directions. Thus, our symmetric MTAA is capable of extracting shared knowledge and predict the correct labels for all tasks.

V. RELATED WORK

Multi-task Learning is a popular approach in different NLP field [14], [15]. For the sequence labeling task, Collobert and Weston (2008) [49] firstly proposed a unified sequence labeling architecture, which applied to various tasks such as SRL, NER, POS tagging, and chunking simultaneously. Søgaard and Goldberg (2016) [17] presented a multi-task learning architecture with deep bi-directional RNNs, where different tasks supervision can happen at different layers. They used the POS tagging as the source task, while the Chunking and CCG supertagging are the target task. They also showed that POS benefits Chunking and CCG. The multi-task relationship, however, they refer to in their work is the source-target relationship. Besides, their method is unscalable and cannot accomplish more than two sequence labeling tasks simultaneously. Moreover, Hashimoto et al. (2017) [25] presented an end-to-end model that can be trained for POS tagging, chunking, dependency parsing, semantic relatedness, and textual entailment. They exploited the linguistic hierarchy structure that treats different tasks at different layers. The main limitation of the linguistic hierarchy structure-based methods is that we have to sort the tasks, while the order of tasks may impact the performance of all tasks. Based on the symmetric structure, Yang et al. (2017) [16] attempted transfer learning for low-resource neural sequence taggers. They proposed three transfer models for cross-domain, cross-application, and cross-lingual transfer for the sequence labeling tasks. Clark et al. 2018 [48] proposed a self-training algorithm, called Cross-View Training (CVT), for the neural sequence model, which leverages both labeled and unlabeled data to improve the representations of sentence encoder. CVT also can easily be combined with multi-task learning. Lu et al. [28] proposed a new LSTM cell, called Shared-Cell LSTM (SC-LSTM), which can learn task-shared and specific information simultaneously. As a result, this new LSTM improved the performance of multi-task sequence labeling. All three methods use a similar symmetric structure to our proposed MTAA. However, none of them consider eliminating task-specific noise from the model.

Adversarial networks also have drawn wide attention in the NLP field [21], [22], [50]. For the Chinese NER, Yang et al. (2018) [20] proposed a method for crowd annotation learning, which can exploit the noisy sequence labels from multiple annotators. Moreover, the authors created two data sets for Chinese NER tasks in the dialog and e-commerce domains. The experimental results show that the proposed approach can surpass strong baseline systems. Moreover, Cao et al. (2018) [19] proposed an adversarial transfer learning framework for the Chinese NER task, which can make full use of task-shared boundaries information and prevent the task-specific features of CWS. Experimental results on two different widely used data sets show that this model significantly and consistently outperforms other state-of-the-art methods. However, as we illustrate in Section II, their methods might suffer from the discriminator collapse problem. Both methods exploited the adversarial training to transfer knowledge from the external corpus, but they only completed one task (e.g., the Chinese NER) limited by the structure. For other NLP tasks, Chenet al. (2017) [29] proposed adversarial multi-criteria learning for CWS by integrating shared

knowledge from multiple heterogeneous segmentation criteria. Compared to single-criterion learning, their model obtained significant improvements on eight corpora with heterogeneous segmentation criteria. Yasunaga et al. (2018) [51] proposed a neural POS tagging model that aims to achieve robustness to input perturbations. Multilingual experiments showed that the proposed model can achieve significant improvements in all tested languages, especially in low resource ones. Chen et al. (2018) [52] proposed an adversarial deep averaging network to tackle the sentiment classification problem in low-resource languages without adequate annotated data.

Recently, attention mechanism has achieved great success in natural language tasks [53], [54], [55]. For machine translation, attention mechanisms establish dependencies regardless of their distance in the input or output sequence [56]. To solve the problem of police killing recognition, Nguyen and Nguyen (2018) [57] introduced supervised attention mechanisms based on semantical word lists and dependency trees to weight the important contextual words. Lin et al. (2017) [58] presented a multi-lingual neural relation extraction framework employing mono-lingual attention and cross-lingual attention. They exploited the two kinds of attention mechanisms to capture the pattern consistency and complementarity among languages.

VI. Conclusions

In this paper, we propose a multi-task learning method MTAA for sequence labeling tasks. MTAA has a symmetric structure that can treat all tasks equally. Our model exploits the adversarial training strategy to purely extract shared knowledge among tasks. Furthermore, the multi-representation fusion attention mechanism generates the fusion representations from shared and individual representations appropriately. Experiments on two well-known data sets show that MTAA achieves significant improvements over the previous state-of-the-art models. For future work, we will further model multi-task sequence labeling without multi-way parallel data, as well as generalize our MTAA to NLP tasks other than the sequence labeling tasks.

References

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[2] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *NAACL HLT*, 2016, pp. 260–270.

[3] J. D. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001, pp. 282–289.

[4] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, "Character-based lstm-crf with radical-level features for chinese named entity recognition," in *Natural Language Understanding and Intelligent Applications*, 2016, pp. 239–250.

[5] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *ACL*, 2016, pp. 1064–1074.

[6] B. Y. Lin, F. Xu, Z. Luo, and K. Zhu, "Multi-channel bilstm-crf model for emerging named entity recognition in social media," in *WNUT*, 2017, pp. 160–165.

[7] Y. Wang, Y. Li, Z. Zhu, B. Xia, and Z. Liu, "Sc-ner: A sequence-to-sequence model with sentence classification for named entity recognition," in *PAKDD*, 2019, pp. 198–209.

[8] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT*, 2019, pp. 4171–4186.

[11] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[12] A. Ghaddar and P. Langlais, "Robust lexical features for improved neural network named-entity recognition," in *COLING*, 2018, pp. 1896–1907.

[13] X. Feng, X. Feng, B. Qin, Z. Feng, and T. Liu, "Improving low resource named entity recognition using cross-lingual knowledge transfer." in *IJCAI*, 2018, pp. 4071–4077.

[14] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," *arXiv preprint arXiv:1603.06270*, 2016.

[15] N. Peng and M. Dredze, "Multi-task domain adaptation for sequence tagging," in *RepL4NLP*, 2017, pp. 91–100.

[16] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," in *ICLR*, 2017.

[17] A. Søgaard and Y. Goldberg, "Deep multi-task learning with low level tasks supervised at lower layers," in *ACL*, 2016, pp. 231–235.

[18] Y. Lin, S. Yang, V. Stoyanov, and H. Ji, "A multi-lingual multi-task architecture for low-resource sequence labeling," in *ACL*, 2018, pp. 799–809.

[19] P. Cao, Y. Chen, K. Liu, J. Zhao, and S. Liu, "Adversarial transfer learning for chinese named entity recognition with self-attention mechanism," in *EMNLP*, 2018, pp. 182–192.

[20] Y. Yang, M. Zhang, W. Chen, W. Zhang, H. Wang, and M. Zhang, "Adversarial learning for chinese ner from crowd annotations," in *AAAI*, 2018.

[21] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *ICLR*, 2016.

[22] P. Liu, X. Qiu, and X.-J. Huang, "Adversarial multi-task learning for text classification," in *ACL*, 2017, pp. 1–10.

[23] X. Wang, X. Han, Y. Lin, Z. Liu, and M. Sun, "Adversarial multi-lingual neural relation extraction," in *COLING*, 2018, pp. 1156–1166.

[24] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.

[25] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, "A joint many-task model: Growing a neural network for multiple nlp tasks," in *EMNLP*, 2017, pp. 1923–1933.

[26] S. Changpinyo, H. Hu, and F. Sha, "Multi-task learning for sequence tagging: An empirical study," in *COLING*, 2018, pp. 2965–2977.

[27] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, "Empower sequence labeling with task-aware neural language model," in *AAAI*, 2018, pp. 5253–5260.

[28] P. Lu, T. Bai, and P. Langlais, "Sc-lstm: Learning task-specific representations in multi-task learning for sequence labeling," in *NAACL HLT*, 2019, pp. 2396–2406.

[29] X. Chen, Z. Shi, X. Qiu, and X. Huang, "Adversarial multi-criteria learning for chinese word segmentation," in *ACL*, 2017, pp. 1193–1203.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 6000–6010.

[31] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[32] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[33] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *NeurIPS*, 2016, pp. 343–351.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[35] E. F. T. K. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *CoNLL*, 2003.

[36] S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: A unified relational semantic representation," in *ICSC*, 2007, pp. 517–526.

[37] W. Daelemans, J. Zavrel, A. van den Bosch, and K. van der Sloot, "Mbt: Memory based tagger, version 1.0, reference guide," *ILK Technical Report*, vol. 2, 2002.

[38] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes," in *EMNLP-CoNLL Shared Task*, 2012, pp. 1–40.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.

[40] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *ICML*, 2013, pp. 1310–1318.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[42] J. Yang and Y. Zhang, "Ncrf++: An open-source neural sequence labeling toolkit," in *ACL*, 2018, pp. 74–79.

[43] M. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," in *ACL*, 2017, pp. 1756–1765.

[44] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING*, 2018, pp. 1638–1649.

[45] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL HLT*, 2018, pp. 2227–2237.

[46] J. Yang, Z. Teng, M. Zhang, and Y. Zhang, "Combining discrete and neural features for sequence labeling," in *CICLING*, 2016, pp. 140–154.

[47] Y. Zhang and J. Yang, "Chinese ner using lattice lstm," in *ACL*, 2018, pp. 1554–1564.

[48] K. Clark, M.-T. Luong, C. D. Manning, and Q. Le, "Semi-supervised sequence modeling with cross-view training," in *EMNLP*, 2018, pp. 1914–1925.

[49] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *ICML*, 2008, pp. 160–167.

[50] J.-K. Kim, Y.-B. Kim, R. Sarikaya, and E. Fosler-Lussier, "Cross-lingual transfer learning for pos tagging without cross-lingual resources," in *EMNLP*, 2017, pp. 2832–2838.

[51] M. Yasunaga, J. Kasai, and D. Radev, "Robust multilingual part-of-speech tagging via adversarial training," in *NAACL HLT*, 2018, pp. 976–986.

[52] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 557–570, 2018.

[53] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *ICLR*, 2019.

[54] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, 2019.

[55] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *NeurIPS*, 2019, pp. 5753–5763.

[56] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[57] M. Nguyen and T. Nguyen, "Who is killed by police: Introducing supervised attention for hierarchical lstms," in *COLING*, 2018, pp. 2277–2287.

[58] Y. Lin, Z. Liu, and M. Sun, "Neural relation extraction with multi-lingual attention," in *ACL*, 2017, pp. 34–43.

Yu Wang is currently a Ph.D. student in the Department of Computer Science, Nanjing University of Posts and Telecommunications, China. He has joined Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing, China. His research interests include sequence labeling and information extraction.

Yun Li received the Ph.D. degree in Computer Science from Chongqing University, Chongqing, China. He is a professor in the School of Computer Science, Nanjing University of Posts and Telecommunications, China. Prior to that, he was the postdoctoral fellow in Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He is the Principal Investigator (PI) of several national scientific research projects and provincial projects in recent years. His research mainly focuses on machine learning, data mining and parallel computing. He has published more than 60 refereed research papers in AAAI, IJCAI, ECML, IEEE Trans. Neural Networks and Learning Systems, Pattern Recognition, etc. He is the member of IEEE.

Ziye Zhu is currently working toward the Ph.D. degree in the Department of Computer Science, Nanjing University of Posts and Telecommunications, China. She has joined Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing, China. Her research interests include software mining and natural language processing.

Hanghang Tong is currently an associate professor at Department of Computer Science at University of Illinois at Urbana-Champaign. Before that he was an associate professor at School of Computing, Informatics, and Decision Systems Engineering (CIDSE), Arizona State University. He received his M.Sc. and Ph.D. degrees from Carnegie Mellon University in 2008 and 2009, both in Machine Learning. His research interest is in large scale data mining for graphs and multimedia. He has received several awards, including IEEE ICDM Tao Li award (2019), SDM/IBM Early Career Data Mining Research award (2018), NSF CAREER award (2017), ICDM 10-Year Highest Impact Paper award (2015), four best paper awards (TUP'14, CIKM'12, SDM'08, ICDM'06), seven 'bests of conference', 1 best demo, honorable mention (SIGMOD'17), and 1 best demo candidate, second place (CIKM'17). He has published over 100 refereed articles. He is the Editor-in-Chief of SIGKDD Explorations (ACM), an action editor of Data Mining and Knowledge Discovery (Springer), and an associate editor of ACM Computing Surveys (ACM), Knowledge and Information Systems (Springer) and Neurocomputing Journal (Elsevier); and has served as a program committee member in multiple data mining, database and artificial intelligence venues (e.g., SIGKDD, SIGMOD, AAAI, WWW, CIKM, etc.).

Yue Huang is currently a Ph.D. student in the Department of Computer Science, Nanjing University of Posts and Telecommunications, China. Before that, he received the master's degree in computer science from Nanjing University of Aeronautics and Astronautics in 2012. His research interests include natural language processing and event summarization.