Improving Low-Resource Chinese Event Detection with Multi-Task Learning

Meihan Tong$^{1,2[0000-0003-2679-5641]}$,
Bin Xu$^{1,2[0000-0003-3040-4391]\star}$, Shuai Wang$^{3[0000-0001-5209-9186]}$,
Lei Hou$^{1,2[0000-0002-8907-3526]}$, and Juaizi Li$^{1,2[0000-0002-6244-0664]}$

[1] Department of Computer Science and Technology, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[2] Knowledge Intelligence Research Center, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China
[3] SLP Group, AI Technology Department, JOYY Inc, China
tongmeihan@gmail.com, xubin@tsinghua.edu.cn
wangshuai1@yy.com, houlei@tsinghua.edu.cn,
lijuanzi@tsinghua.edu.cn

Abstract. Chinese Event Detection (CED) aims to detect events from unstructured sentences. Due to the difficulty of labeling event detection datasets, previous approaches suffer from severe data sparsity problem. To address this issue, we propose a novel Lattice LSTM based multi-task learning model. On one hand, we utilize multi-granularity word information via Lattice LSTM to fully exploit existing datasets. On the other hand, we employ the multi-task learning mechanism to improve CED with datasets from other tasks. Specifically, we combine Name Entity Recognition (NER) and Mask Word Prediction (MWP) as two auxiliary tasks to learn both entity and general language information. Experiments show that our approach outperforms the six SOTA methods by 1.9% on ACE2005 benchmark. We will release the code once the paper is accepted.

Keywords: Chinese Event Detection · Multi-task Learning · Lattice LSTM

1 Introduction

Chinese Event Detection (CED) aims to extract event triggers from sentences [3]. As illustrated in Figure 1, CED needs to identify word “卸下” (resigned) in S_1 as the event trigger for *Resign* event. CED supports a large number of applications , and benefits various downstream tasks such as event knowledge graph building [16].

Mainstream methods can be divided into two categories: feature-based and neural-based models. Feature-based models [6, 15] build lexical features and sentence features with NLP tools to detect event. Neural-based models adopt dynamic pooling CNN [2], attention enhanced Bi-LSTM [23] or Nugget Proposal Network [8] to automatically obtain useful features to improve CED.

Although previous methods have achieved great success, the performance of these supervised methods is largely limited by the amount of annotated data. Namely, labeled

* Corresponding author.

data sparsity is the bottleneck for CED [11]. Benchmark datasets for CED are small: 697 articles for ACE2005[4]. Due to the complexity of CED, it is time-consuming and labor-intensive to manually label more data. Some distant supervision methods [20] have been proposed to automatically augment datasets with existing knowledge bases. However, due to the ambiguity of event trigger (illustrated in Figure 1), corpora labeled in this way are of low quality and insufficient coverage. In addition, compared with English, Chinese knowledge bases are much smaller [18], which limits the ability of distant supervision methods to address the data sparsity of CED.

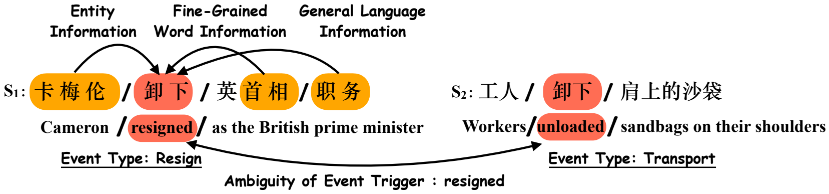
Entity Information
Fine-Grained Word Information
General Language Information
S1: 卡梅伦 / 卸下 / 英首相 / 职务
Cameron / resigned / as the British prime minister
Event Type: Resign
S2: 工人 / 卸下 / 肩上的沙袋
Workers / unloaded / sandbags on their shoulders
Event Type: Transport
Ambiguity of Event Trigger : resigned

Fig. 1. An instance of CED

We address the data sparsity issue from two aspects. On one hand, we delve into the characteristics of Chinese to utilize textual information in more detail. On the other hand, we introduce the idea of multi-task learning to leverage the annotated data from other tasks to improve CED. We illustrate the details as follows.

Chinese is a non-delimiter language, so Chinese words have different granularity and sub-word can also be a word. For instance, “英首相” (British Prime Minister) can be considered as a coarse-grained word, or can be disassembled as two fine-grained words “英” (British) and “首相” (Prime Minister). Previous methods totally ignore the fine-grained word information, which makes them unable to leverage the fine-grained words information to understand the semantics of coarse-grained words. As illustrated in Figure 1), coarse-grained word “英首相” (British Prime Minister) is a rare word whose semantic is difficult to learn, but with the semantic support of the fined-grained word “首相” (Prime Minister) which means the head of the government, we have more confidence to say that “卸下” (resigned) triggers a *Resign* event instead of a *Transport* event. In our architecture, we employ Lattice LSTM to simultaneously capture word information of different granularity.

For the multi-task learning, we notice that named entity has a great impact on event classification [10]. In S_1, knowing the entity type of “卡梅伦” (Cameron) is a politician, we have more evidence to infer that “卸下” (resigned) triggers a *Resign* event. Fortunately, as a well-studied task, Named Entity Recognition (NER) has large-scale annotated corpora, making it convenient to learn entity information by employing NER as an auxiliary task. We also notice that general language information obtained from large-scale plain data is helpful to disambiguate the event trigger. Recently, by pre-training on large-scale corpora, many works [5] have achieved state-of-the-art perfor-

[4] https://catalog.ldc.upenn.edu/LDC2006T06

mance on multiple information extraction tasks. Following these works, we design a variant of language model called Mask Word Prediction (MWP) to learn general language information in plain text.

In this paper, we propose a novel **M**ulti-**L**earning enhanced **L**attice LSTM model called MLL to address the data sparsity issue of CED. We formula CED as a sequence labeling task. Specifically, we exploit Lattice LSTM to encode multi-granularity word information by adding extra paths to traditional LSTM cell, and employ NER and MWP to leverage entity and general language information to eliminate event ambiguity. CED, NER and MWP share the parameters of feature extraction layer, and then we adopt three different matrices to map them to respective CRF layers for sequence labeling. Finally, we evaluate our model on benchmark ACE2005. Experiments show that our model outperforms six SOTA methods.

Our contributions can be summarized as: 1) To the best of our knowledge, we propose a novel MLL model for Chinese Event Detection, which leverages multi-granularity word information to eliminate event trigger ambiguity, and successfully utilize such information via a bi-directional Lattice LSTM. 2) Instead of only utilizing CED annotated corpus, we innovatively leverage multi-task learning to capture entity information from entity-rich corpus and general language information from plain text to improve CED. Experiment results show that the two auxiliary tasks NER and MWP complement each other by improving the precision and recall respectively. 3) We surpass six SOTA methods on benchmark ACE2005, which raises the F1-value by 1.9%.

2 Problem Definition

Our multi-task learning model involves three tasks: Chinese Event Detection (CED) and two auxiliary tasks, i.e., Named Entity Recognition (NER) and Mask Word Prediction (MWP). We first define the common symbols shared by the three tasks and then introduce the task-specific symbols.

Given a sentence $S = \langle x_1, x_2, \dots, x_n \rangle$, where x_j stands for the j-th character and $x_{k:j}$ represents the word composed by character sequence $\langle x_k, x_{k+1}, \dots, x_j \rangle$. For instance, in S1, x_7 refers to character “首” (prime) and $x_{7:8}$ refers to word “首相” (Prime Minister). Since Chinese words have various granularity, different words can end with the same character. We denote $X_j = \{x_{k_1:j}, x_{k_2:j}, \dots, x_{k_l:j}\}$ as the collection of words ending with the j-th character x_j. $I_j = \{k_1, k_2, \dots, k_l\}$ records the start position of these words. For example, in S1, both “英首相” (British Prime Minister) and “首相” (Prime Minister) are end with character “相” (Minister), so X_8={英首相(British Prime Minister), 首相(Prime Minister)} and I_8={6,7}. $X = \bigcup_{i=1}^{n} X_i$ represents all of the words detected from sentence S.

According to the definition of Automatic Content Extraction (ACE) [3], **Chinese Event Detection (CED)** aims to identify event trigger from sentences. Formally, given a sentence $S_c = \langle x_1, x_2, \dots, x_n \rangle$ and word collection X_c, CED task aims to determine the event type y_j triggered by the j-th character x_j in S_c. Event type label $y_i \in \{\textit{Resign}, \textit{Injure}, \dots, \textit{Die}\}$ refers to the 33 predefined event classes in ACE system. $Y_c = \langle y_1, y_2, \dots, y_n \rangle$ denotes the event trigger label sequence for S_c. **Named Entity Recognition (NER)** aims to identify whether the i-th character x_i in sentence

S_n is a named entity (name of people, organization, places, etc.). For both CED and NER, we adopt BMES mechanism to label event trigger and named entity. **Mask Word Prediction (MWP)** is a whole-word masking task. At the input side, we randomly replace some characters with symbol “MASK”. At the output side, we desire the model to predict the masked characters. Formally, given a sentence S_m that lacks the i-th word, MWP needs to find the original word x_i.

Essentially, given the sentence S_c, S_n, S_m and its annotated labels Y_c, Y_n, Y_m from CED, NER, MWP tasks respectively, our model estimates the sum of the probabilities of $P_c(Y_c|S_c)$, $P_n(Y_n|S_n)$ and $P_m(Y_m|S_m)$.

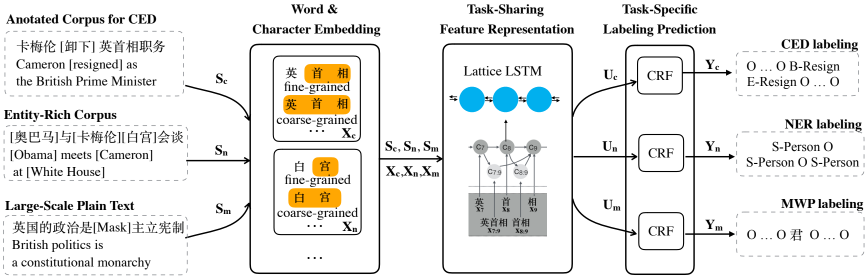

Fig. 2. The overall architecture of MLL. From left to right is Word and Character Embedding, Task-Sharing Semantic Representation and Task-Specific Label Prediction. All tasks share the first two layers, and the last layer is specific for each task.

3 Methodology

We propose a **M**ulti-task **L**earning enhanced **L**attice LSTM model, called MLL, to handle data sparsity issue in CED. In this section, we first introduce the architecture of MLL (Figure 2), and then introduce each component in detail. MLL consists of three modules. **Word and Character Embedding** detects multi-granularity words from sentence (like “首相” (Prime Minister),“英首相” (British Prime Minister) in S1) and jointly transforms characters (like “英” (British),“首” (Prime),“相” (Minister)) and multi-granularity words into the same semantic embedding space. **Task-Sharing Semantic Representation** deploys bi-directional Lattice LSTM to simultaneously encode character sequences information and multi-granularity word sequences information. Comparing with traditional LSTM cell which only accepts a single input, the input to Lattice LSTM cell can be a collection of multi-granularity words that end with the same character. In S1, the input of the character “相” (Minister) not only comes from character “首” (Prime), but also from overlapping word “首相” (Prime Minister) and “英首相” (British Prime Minister). **Task-Specific Labeling Prediction** transforms the task-sharing features into task-specific features, and then deploys CRF to find the optimal labeling sequence for each task. The parameters of Word and Character Embedding and Task-Sharing Semantic Representation are shared by three tasks, while the parameters of Task-Specific Labeling Prediction are task-specific.

3.1 Word and Character Embedding

In this section, we generate word collection X from sentence S. We first construct a large word dictionary, and then look up multi-granularity words from sentences, including the overlapping ones.

To construct a large word dictionary, we first employ three word segmentation tools (NLPIR, THULAC and LTP) to split sentences in Gigaword corpus. In this way, we obtain different word sequences for the same sentence. Then we filter out low-frequency words. Since the word dictionary is large (containing 5.7k Chinese characters and 698.7k words), we exploit Trie Dictionary algorithm to improve query efficiency.

After detecting word list X_j={$x_{k_1:j}$, $x_{k_2:j}$, $\ldots$, $x_{k_l:j}$} for each character x_j, we represent characters and words with pre-trained embedding. We obtain the pre-trained embedding following [21], which adopts word2vec as model. $\mathbf{x_j}$ and $\mathbf{x_{k:j}}$ refer to the embedding representation for character x_j and word $x_{k:j}$.

3.2 Task-Sharing Feature Representation

Traditional LSTM cell can receive only one input, word embedding or character embedding. Therefore, previous LSTM-based models can only utilize word information from one specific segmentation. However, Lattice LSTM [22] can accept inputs of different lengths, which is especially useful for language without natural word separator like Chinese. With multi-input Lattice LSTM cell (shown in Figure 3), our model is able to consider words with different granularity, which provide richer sentence semantics.

Formally, at the j-th step, the input of Lattice LSTM is grouped into two categories: the current character $\mathbf{x_j}$ and the word list $\mathbf{X_j} = \{\mathbf{x_{k_1:j}}, \mathbf{x_{k_2:j}}, \ldots, \mathbf{x_{k_l:j}}\}$, where l denotes the number of words ended with the j-th character. Noted that l varies according to the context. For instance in S1, X_7={“英首相” (British Prime Minister), “首相” (Prime Minister)} has a length of 2, X_2={“卡梅伦” (Cameron)} has a length of 1 and X_0={} has a length of 0.

Character Representation Standard LSTM is used to encode the current character $\mathbf{x_j}$.

$$\begin{aligned} f_j &= \sigma(W_a[\mathbf{x_j}; h_{j-1}] + b) \\ i_j &= \sigma(W_a[\mathbf{x_j}; h_{j-1}] + b) \\ o_j &= \sigma(W_a[\mathbf{x_j}; h_{j-1}] + b) \\ \tilde{c}_j &= \tanh(W_a[\mathbf{x_j}; h_{j-1}] + b) \end{aligned} \tag{1}$$

where f_j, i_j and o_j are the forget gate, input gate and output gate for character $\mathbf{x_j}$, and h_{j-1} is the final hidden representation of previous character $\mathbf{x_{j-1}}$. “;” represents the concatenation operation.

Word Representation For each word $\mathbf{x_{k:j}} \in \mathbf{X_j}$, we obtain its cell state $c_{k:j}$ by deploying a variant of LSTM cell.

$$\begin{aligned} f_{k:j} &= \sigma(W_b[\mathbf{x_{k:j}} + b) \\ i_{k:j} &= \sigma(W_b[\mathbf{x_{k:j}}; h_{j-1}] + b) \\ \tilde{c}_{k:j} &= \tanh(W_b[\mathbf{x_{k:j}}; h_{j-1}] + b) \\ c_{k:j} &= f_{k:j} \cdot c_{j-1} + i_{k:j} \cdot \tilde{c}_{k:j} \end{aligned} \tag{2}$$

where c_{j-1} is the final cell representation of previous character $\mathbf{x_{j-1}}$.

In the same way, we calculate cell state $c_{k:j}$ for each $\mathbf{x_{k:j}} \in \mathbf{X_j}$. Finally, we can obtain the collection of cell states $C_j = \{c_{k_1:j}, c_{k_2:j}, \ldots, c_{k_l:j}\}$.

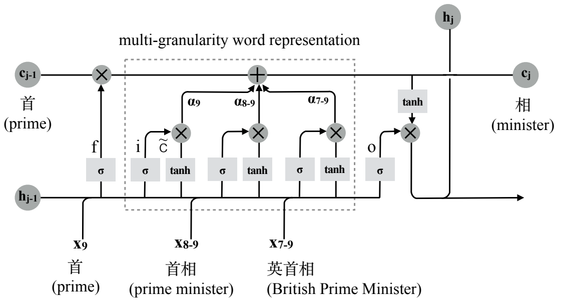

Fig. 3. The details of multi-granularity word information representation.

Hybrid Representation As illustrated in Figure 3, we aggregate character representation $\tilde{c}_j$ and word representation $C_j = \{c_{k_1:j}, c_{k_2:j}, \ldots c_{k_l:j}\}$ by self-attention mechanism. Attention weights are calculated by the following equations.

$$\begin{aligned} \alpha_j &= \frac{\exp i_j}{\exp i_j + \sum_{k' \in I_j} \exp i_{k':j}} \\ \alpha_{k:j} &= \frac{\exp i_{k:j}}{\exp i_j + \sum_{k' \in I_j} \exp i_{k':j}} \end{aligned} \tag{3}$$

where $I_j = \{k_1, k_2, \ldots, k_l\}$ indicates the start index of word list $\mathbf{X_j}$.

Finally, we can calculate the cell representation c_j and hidden representation h_j for the j-th character $\mathbf{x_j}$ by

$$\begin{aligned} c_j &= \alpha_j \cdot \tilde{c}_j + \sum_{k \in I_j} \alpha_{k:j} \cdot c_{k:j} \\ h_j &= o_j \cdot \tanh(c_j) \end{aligned} \tag{4}$$

where o_j refers to the output gate representation of character $\mathbf{x_j}$ defined in Eq.1.

Now, we can obtain the task-sharing feature H by concatenating hidden representation h_j. In order to capture both the forward and backward information, we adopt the Bi-directional Lattice LSTM model.

$$\begin{aligned} h_i &= [\overrightarrow{h_j}; \overleftarrow{h_j}] \\ H &= [h_1; h_2; \ldots; h_n] \end{aligned} \tag{5}$$

3.3 Task-Specific Labeling Prediction

CED Output: For mainline task Chinese event detection, we adopt a fully connected layer to map the task-sharing feature H into the task-specific representation U_c by

$$U_c = W_c \cdot H + b_c \tag{6}$$

Then a CRF layer is used to predict the event trigger label based on task-specific feature $U_c = [u_1; u_2; \ldots; u_n]$. The probability of a label sequence $P_c(Y_c|S_c)$ is

$$P_c(Y_c|S_c) = \frac{\exp \sum_j^n (W_{y_j} u_j + b_{y_{j-1}:y_j})}{\sum\limits_{Y_c \in E(Y_c)} \exp \sum_j^n (W_{y_j} u_j + b_{y_{j-1}:y_j})} \tag{7}$$

where S_c is the training sentence of Chinese event detection task, Y_c stands for the labels of the event type, E(Y_c) is the full permutation of Y_c.

NER and MWP Output: Now, we transform task-sharing feature H to task-specific representations U_n and U_m for two auxiliary tasks NER and MWP. Analogous to the calculation formulas of U_c (defined in Eq.6) and $P_c(Y_c|S_c)$ (defined in Eq.7), we employ fully-connected layers and CRF layers to calculate U_n, $P_n(Y_n|S_n)$ for NER task and U_m, $P_m(Y_m|S_m)$ for MWP task respectively.

3.4 Joint Training

In this section, we introduce the details for training three tasks simultaneously. Instead of alternate training, we accumulate the loss from three tasks and jointly optimize the model parameters. In this way, the parameters from Task-Sharing Feature Representation will be updated with regard to all the three tasks, which helps the model to discover features that are beneficial for all the three tasks and prevent the model from over-fitting one specific task. The final loss function is:

$$\begin{aligned} L(\theta) = -(&\sum_{i=1}^{N_c} log(P_c(Y_{ci}|S_{ci})) + \sum_{i=1}^{N_n} log(P_n(Y_{ni}|S_{ni})) + \\ &\sum_{i=1}^{N_m} log(P_m(Y_{mi}|S_{mi})) + \frac{\lambda}{2}||\theta||^2) \end{aligned} \tag{8}$$

where N_c, N_n, N_m is the corpus length for CED, NER, MWP task respectively, θ is the parameter sets and λ is the weight of the L2 regulation.

4 Experiment

In this section, we evaluate MLL on widely-used benchmark ACE2005. We will first introduce the experiment settings, then demonstrate the effectiveness of our model by comparing with several baselines, and finally investigate whether the model successfully addresses the data sparsity problem.

4.1 Experimental Settings

Datasets. For our mainline task Chinese event detection task, we use ACE2005, which has 33 predefined event types respectively. Following [21], we split ACE2005 into training, validate and test sets with each having 569/64/64 articles. We utilize NLTK to separate articles into sentences. For auxiliary task NER, we adopt MSRA as benchmark

Model	Trigger Identification			Trigger Classification		
	P	*R*	*F*	*P*	*R*	*F*
Rich-C	58.9	68.1	63.2	58.9	68.1	63.2
DMCNN-C	60.1	61.6	60.9	57.1	58.5	57.8
C-BiLSTM	65.6	66.7	66.1	60.0	60.9	60.4
DMCNN-W	64.1	63.7	63.9	59.9	59.6	59.7
HNN	**74.2**	63.1	68.2	**77.1**	53.1	63.0
NPN	64.8	**73.8**	69.0	60.9	**69.3**	64.8
MLL(our)	**74.2**	69.9	**72.0**	68.7	64.8	**66.7**

Table 1. Overall Performance. The first model is the best feature-based model for ACE2005. C-BiLSTM and HNN are the best character/word-based models. NPN is the current state-of-the-art model. We directly cite the best experiment results from the original papers.

dataset, which contains 39.0k Chinese annotated sentences. For auxiliary task MWP, we build the training corpus from Chinese Wikipedia [5]. We want masked words to be meaningful. Therefore, instead of random choice, we only mask words with outer links in Wikipedia. We filter the corpus w.r.t masked words frequency, and finally obtain 135.9k annotated sentences.

Experiment Detail. All of the three tasks share the same Chinese character and word dictionary. The dimension of character, word and LSTM hidden embedding are 50/50/200. Stochastic gradient descent (SGD) is used for optimization with learning rate as 1.5e-2, decay rate as 5e-2 and momentum as 0.9. The training is conducted on a TitanX GPU. The regularization weight λ and lattice dropout are set to 1e-8 and 0.5 respectively. For comparison, we use precision, recall and F1 score as evaluation metrics following [21].

Baselines. We compare our model with feature-based, word-based and character-based models. **Rich-C [1]** utilizes rich knowledge sources from character to discourse level to detect Chinese event from sentences. **DMCNN [2]** splits the max-pooling layer into two parts according to event trigger position. The inputs to DMCNN-W and DMCNN-C are word sequences and character sequences respectively. **C-BiLSTM [21]** simultaneously considers n-gram information and sequence information at the character level. **HNN [4]** combines Bi-directional LSTM and convolutional neural networks to learn language-independent features for event detection task. **NPN [8]** proposes a nugget proposal network to deal with the word-trigger mismatch problem.

4.2 Overall Performance

Table 1 presents the overall results. We have the following observations. 1) MLL sets a new performance on benchmark ACE2005. Compared with baselines, MLL improves the F1 score by 1.9% on ACE2005 which proves the effectiveness of our model to handle labeled data sparsity issue of Chinese event detection. 2) MLL is consistently superior to feature-based models. Feature-based methods have pipeline error propaganda issue. Their accuracy is limited by the performance of word segmentation and feature

[5] https://dumps.wikimedia.org

extraction tools. 3) MLL significantly increases the recall comparing with character-based and word-based models, proving that jointly considering character, word, named entity and general language information can make MLL (our) more robust in semantic understanding. 4) MLL outperforms strong baseline NPN. NPN adopts two separate dynamic multi-pooling CNN to respectively encode character and word, and then integrates them via attention mechanism. We argue that our model combines character and word information at earlier stage, thus is more powerful in capturing sentences semantics of various granularity.

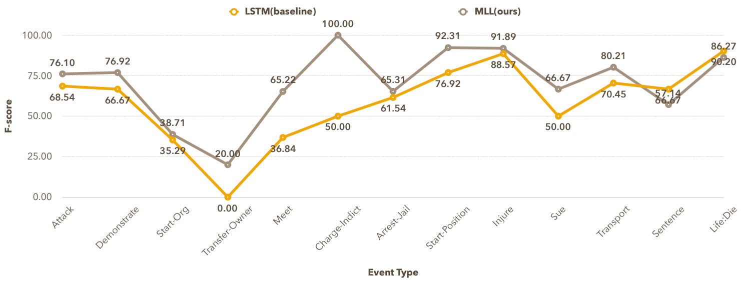

Fig. 4. Analysis from the event type: how and when MLL(ours) helps CED.

It would be interesting to see the improvement on each type and show which type benefits more from these auxiliary tasks. Compared with traditional LSTM (as shown in Figure 4), MLL (ours) gains benefits on most of the event types in ACE2005. Among them, MLL achieves the most improvement on *Charge-Indict*, *Meet* and *Transfer-Owner*. We find that large proportion of the triggers in these event types are unseen during training. We analyse the radio of unseen triggers in these event type, which shows that 38.7%/30.0%/30.0% of the triggers in *Transfer-Owner*/*Charge-Indict*/*Meet* are invisible during training. The discrepancy between training and test data leads to the inferior performance of the data-driven baseline. Our model surpasses the baseline by incorporating the entity information and general language information. Specifically, the entity information brings the co-occurrence and exclusion constraints between entity type and event type, which provides more insight for trigger disambiguation. The general language information prevents the model from overfitting the trigger word, and thus enhance unseen trigger detection.

4.3 Effectiveness of Multi-Granularity Word Information

We remove the multi-task learning part of our model to analyze the effectiveness of Lattice LSTM. We compare our model with character-based model C-BiLSTM(charcter), word-based model C-BiLSTM(word). [21].

As illustrated in Figure 5, Lattice LSTM outperforms character-based, word-based C-LSTM, which proves the effectiveness of multi-granularity word information in event detection. Specifically, 1) Lattice LSTM is superior to character-based model by significantly improving the precision (from 60.0% to 71.4%). Without multi-granularity word

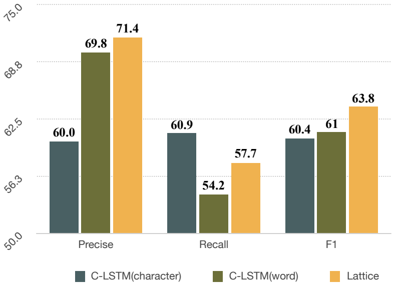

Fig. 5. Multi-granularity word enhancement.

Methods	*P*	*R*	*F*
LSTM	69.9	50.6	58.7
LSTM+NER	71.3	52.6	60.5
LSTM+NER+MWP	69.0	**57.0**	**62.4**
lattice	71.4	57.7	63.8
lattice+NER	77.6	57.7	65.8
lattice+NER+MWP	68.7	**64.8**	**66.7**

Fig. 6. Enhancement from auxiliary task.

information, character-based model has limited ability to disambiguate the event trigger. For instance, without understanding that the word “死胡同” (dead end) means “没有路” (no roads), “死” (dead) can easily be mistaken for the trigger of a “Die” event. 2) Lattice LSTM outperforms word-based model by significantly improving the recall (from 54.2% to 57.7%). Word-based model regards coarse-grained word as basic labeling units, and thus cannot detect event trigger within a word. For instance, since the word “谋杀案” (homicide case) is fed into word-based model as a whole, word-based model cannot mark sub-word “谋杀” (homicide) as the trigger of “Kill” event.

4.4 Effectiveness of Auxiliary Task

We adopt the standard LSTM as task-sharing feature representation layer to prove that our multi-task learning mechanism is still effective without Lattice LSTM.

As illustrated in Table 6, NER auxiliary task consistently improves the precision by 1.4% and 6% for LSTM and Lattice LSTM respectively. We argue that with additional supervision from NER, our model can leverage entity information to eliminate the ambiguity of the event trigger. Considering the example of “卡梅伦黯然离开国会” (Cameron left Congress in dismay), knowing that “卡梅伦” (Cameron) is a politician, we will have more confidence that “离开” (left) is the event trigger for *Resign* rather than *Transport*. Besides, both LSTM + NER + MWP and lattice + NER + MWP are superior to LSTM + NER and lattice + NER by significantly increasing the recall (+1.9/+0.9), which proves the effectiveness of MWP task. A possible reason is that both NER and CED are human-labeled corpus, containing noise patterns that mislead the training. With the aid of MWP, the model is able to learn general language information from large-scale plain corpus, resulting in more robust in semantic understanding.

If we combine the results from Figure 5 and Table 6, we have an interesting discovery: with much higher level semantic meaning captured by model (from character, word to named entity), the precision gradually increases (from 60.0, 69.8 to 77.6). This is consistent with human cognition: higher level of semantic knowledge improves the understanding of the text.

5 Related Work

Event Detection. Numerous methods have been adopted to handle Chinese event detection. We divide them into two categories: feature-based model and neural-based model. **Feature-based models** adopt manually constructed features to classify the event triggers. These features include lexical features [1] (such as n-gram sequence, part-of-speech tags, named entity), sentence-level features [14] (such as syntactic feature, semantic role labeling, event argument labeling) and document-level features [7]. **Neural-based models** exploit end-to-end neural networks to improve CED. DMCNN adopts a dynamically multi-pooling CNN [2, 13] and others employ the hybrid CNN/LSTM model to leverage n-gram and sequential information to detect event [21, 4, 9]. [23] improves event detection via document information. However, previous approaches suffer from data sparsity issue. We address it via Multi-task Learning model. Our model MLL jointly exploits rich NER resources and abundant unlabeled resources to improve CED.

Multi-Task Learning. Multi-task learning has achieved remarkable results on widely NLP tasks such as domain-specific NER [17] and machine translation of low-resource language [19]. [12] is the most relevant work, which adopts Word Sense Disambiguation(WSD) as auxiliary task to improve event detection. However, WSD has two drawbacks. Firstly, the training corpus of WSD is hard to obtain, while our auxiliary tasks NER and MWP either have a large number of existing annotation corpora or do not need manual annotation. Secondly, it does not employ language model task, so general language information cannot be utilized.

6 Conclusion

We propose a **L**attice-LSTM based **M**ulti-task **L**earning model to address the data sparsity issue of Chinese event detection. Specifically, we employ Lattice LSTM to capture multi-granularity word information to fully exploit exiting datasets, and leverage two auxiliary tasks NER and MWP to capture named entity and general language information to utilize datasets from other tasks. Experiment demonstrates that MLL outperforms traditional methods and achieves state-of-the-art performance on benchmark ACE2005.

7 Acknowledgments

This work is supported by the National Key Research and Development Program of China (2018YFB1005100 and 2018YFB1005101), NSFC key projects (U1736204, 61533018), and grants from Beijing Academy of Artificial Intelligence (BAAI2019ZD0502) and the Institute for Guo Qiang, Tsinghua University (2019GQB0003). It also got partial support from National Engineering Laboratory for Cyberlearning and Intelligent Technology, and Beijing Key Lab of Networked Multimedia.

References

1. Chen, C., Ng, V.: Joint modeling for chinese event extraction with rich linguistic features. COLING pp. 529–544 (2012)

2. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: ACL. vol. 1, pp. 167–176 (2015)
3. Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S.M., Weischedel, R.M.: The automatic content extraction (ace) program-tasks, data, and evaluation. In: LREC (2004)
4. Feng, X., Qin, B., Liu, T.: A language-independent neural network for event detection. Science China Information Sciences **61**(9), 092106 (2018)
5. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
6. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: ACL (2013)
7. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: ACL. pp. 789–797 (2010)
8. Lin, H., Lu, Y., Han, X., Sun, L.: Nugget proposal networks for chinese event detection. arXiv preprint arXiv:1805.00249 (2018)
9. Lin, H., Lu, Y., Han, X., Sun, L.: Cost-sensitive regularization for label confusion-aware event detection. arXiv preprint arXiv:1906.06003 (2019)
10. Liu, J., Chen, Y., Liu, K.: Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In: AAAI. pp. 6754–6761 (2019)
11. Liu, S., Chen, Y., He, S., Liu, K., Zhao, J.: Leveraging framenet to improve automatic event detection. In: ACL. vol. 1, pp. 2134–2143 (2016)
12. Lu, W., Nguyen, T.H.: Similar but not the same: Word sense disambiguation improves event detection via neural representation matching. In: EMNLP (2018)
13. Makarov, P., Clematide, S.: Uzh at tac kbp 2017: Event nugget detection via joint learning with softmax-margin objective. In: TAC (2017)
14. McClosky, D., Surdeanu, M., Manning, C.D.: Event extraction as dependency parsing. In: ACL. pp. 1626–1635 (2011)
15. Miwa, M., Sætre, R., Kim, J.D., Tsujii, J.: Event extraction with complex event classification using rich features. Journal of bioinformatics and computational biology (2010)
16. Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., Soroa, A., Ploeger, T., Bogaard, T.: Building event-centric knowledge graphs from news. Journal of Web Semantics (2016)
17. Wang, X., Zhang, Y., Ren, X., Zhang, Y., Zitnik, M., Shang, J., Langlotz, C., Han, J.: Cross-type biomedical named entity recognition with deep multi-task learning. arXiv preprint arXiv:1801.09851 (2018)
18. Yang, T.H., Huang, H.H., Yen, A.Z., Chen, H.H.: Transfer of frames from english framenet to construct chinese framenet: A bilingual corpus-based approach. In: LREC (2018)
19. Zaremoodi, P., Buntine, W., Haffari, G.: Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In: ACL (2018)
20. Zeng, Y., Feng, Y., Ma, R., Wang, Z., Yan, R., Shi, C., Zhao, D.: Scale up event extraction learning via automatic training data generation. In: AAAI (2018)
21. Zeng, Y., Yang, H., Feng, Y., Wang, Z., Zhao, D.: A convolution bilstm neural network model for chinese event extraction. In: NLPCC. Springer (2016)
22. Zhang, Y., Yang, J.: Chinese ner using lattice lstm. arXiv preprint arXiv:1805.02023 (2018), `https://arxiv.org/pdf/1805.02023`
23. Zhao, Y., Jin, X., Wang, Y., Cheng, X.: Document embedding enhanced event detection with hierarchical and supervised attention. In: ACL. vol. 2, pp. 414–419 (2018)