**\section{Experiments}**

**\subsection{Experimental Setup}**

**Dataset and evaluation methodology**

We used the standard ACE 2005 corpus in our evaluation, which contains 633 Chinese documents. Unlike English, the ACE Chinese corpus does not have a recognized partition of documents for evaluation. Most of the previous work (\*\*\*) randomly selects 10% of the 633 documents as the test set. However, as reported by Chen and NG (\*\*\*) via 10-fold cross validation, performances achieved on different partitions vary considerably due to the small size of test sets. As a result, cross validation is necessary when conducting experiments on the ACE 2005 corpus. In order to make more accurate evaluation and save time, we perform 5-fold cross-validation experiments.

**Evaluation measures**

Similar to previous work, we evaluated our models in terms of $precision (P)$, $recall (R)$, and $F{-}measure (F)$ for each subtask. These performance metrics are computed according to the following standards of correctness for four subtasks:

\begin{itemize}

\item For trigger identification, a trigger is correctly identified if its offsets exactly match a reference trigger;

\item For trigger classification, a trigger is correctly classified if its trigger type and offsets exactly match a reference trigger;

\item For argument identification, an argument is correctly identified if its offset, related trigger type and trigger’s offsets exactly match a reference argument;

\item For argument classification, an argument is correctly classified if its offsets, role, related trigger type and trigger’s offsets exactly match a reference argument.

\end{itemize}

**Our Method vs. State-of-the-art Methods**

Table \*\*\* shows the overall performance of all methods on the ACE2005 Chinese corpus.

We select the following state-of-art methods for comparison.

Char-MEMM is the first character-based method to handle the language specific issue, which trains a Maximum Entropy Markov Model to label each character with BIO tagging scheme.

Rich-L is a joint-learning, knowledge-rich approach that extends the union of the features employed by Char-MEMM and Li et al. with six groups of linguistic features, including character-based features and discourse consistency features, which is the feature-based state-of-art system.

HNN is a hybrid neural network model, which also incorporates both bidirectional LSTMs and convolutional neural networks to capture sentence and structure semantic information, and it achieves state-of-the-art performance in Chinese event detection.

Our reported results are averaged over five folds, and all results of previous work are listed in their paper. Rich-L performs 10-fold cross validation, while Char-MEMM and HNN simply select their test documents randomly. Therefore, it may be unfair for us to directly compare our results with them.

Compared with feature-based methods. 按照任务进行比较 or 按照方法进行比较。

In event detection, our C-BiLSTM model looks the same as HNN model since they both concatenate the output vector of BiLSTM and CNN, but their CNN parts and final outputs are quite different. CNN in C-BiLSTM learns a representation over shallow windows for every word, rather than the entire sentence as in HNN model. We argue that C-BiLSTM can obtain more accurate contextual information, because different words have different context, and as a result, should have different contextual feature representations instead of sharing one representation. Moreover, HNN treats event detection as a classification task, so it can not identify neither inside-word nor cross-word triggers.

**Word-based Model vs. Character-based Model**

As we can summarize from Table, when applying the same network architecture, word-based methods always have higher precisions while character-based methods always have higher recalls. We then take a further step to see their impacts on different kinds of triggers. Table 3 shows that:

1. Both two methods achieve similar F-measure in regular trigger identification;
2. Word-based methods can not label inside-word triggers, while character-based methods can handle this issue nicely, which brings them higher overall recall;
3. It is harder for character-based method to correctly identify cross-word triggers. As there are more cross-word triggers than inside-word triggers in dataset, the overall F-measure of word-based method is slightly higher.
4. We can draw a similar conclusion in argument identification, as it is affected by the performance of upstream trigger labeling in the pipeline system.

We find several reasons that cause the lower precision of character-based method:

1. Character-based method needs to learn word segmentation by itself. 7.3\% of triggers identified by it are partially mislabeled, like triggers in S8 and S9 in Table~\ref{tab:five}.
2. Word embedding brings richer semantic information than character embedding. Take S10 as an example, characters “胡 ” and “同 ” do not have any meaning related to the formed word “胡同 ” (the end of a road), while this word strongly suggests that “死” (dead) is not a trigger. Given the more accurate embedding of surrounding context, word-based networks can understand the meaning of the center word better and do better disambiguation.
3. LSTM in character-based method needs to maintain clues for longer sequence, as 1.7 times longer than the average length of word sequences. Evaluating on sentences containing more than 150 characters, F-measure of character-based method is 70\%, while word-based method can achieve 72.8\%.

**Neural Network Architectures**

To compare the effectiveness of different parts of C-BiLSTM, we detect events by using BiLSTM and CNN separately. As Table~\ref{tab:four} shows, BiLSTM is slightly more efficient than CNN, and the combined C-BiLSTM model outperforms other two models. This observation demonstrates that both of the two models are important for event detection.

Some words can trigger different types of events according to their contexts, like the word \begin{CJK}{UTF8}{gbsn}``成立''\end{CJK} (found) in S11 and S12. These words account for 36.8 percent of all triggers. So we evaluate the capacity of each network on trigger disambiguation.

S11: 部队 向 抗议者 释放 了 催泪弹 。The troops released tear gas to the protesters. [Type: Attack]

S12：他 在 服刑 五 年 后 从 狱中 释放 出来 。He was released from prison after serving a sentence of five years. [Type: Release-Parole]

从Table 6中可以看出，CNN的消歧效果比 BiLSTM 更好，这得益于 CNN 为每个单词抽取的局部信息。同时，BiLSTM 抽取的句子级别信息也有一定的消除歧义的作用。

Table~\ref{tab:six} provides suggestive evidence that, benefited from the salient contextual features, CNN can perform better trigger disambiguation than BiLSTM. And LSTM-extracted sentence-level information also help reduce some errors caused by ambiguous triggers. C-BiLSTM model that combines both of the two-level features achieve highest precision as we expected.

which makes trigger disambiguation an essential and critical

\begin{algorithm}[t]

\SetAlgoNoLine

\ForEach{\tt epoch}{

\ForEach{\tt training instance}{

1) Bidirectional LSTM forward pass: \\

\hspace{0.5cm} forward pass for forward LSTM \\

\hspace{0.5cm} forward pass for backward LSTM \\

2) Forward pass for CNN \\

3) CRF layer forward and backward pass \\

4) Bidirectional LSTM backward pass: \\

\hspace{0.5cm} backward pass for forward LSTM \\

\hspace{0.5cm} backward pass for backward LSTM \\

5) Backward pass for CNN \\

6) Update parameters

}

}\caption{Convolution bidirectional LSTM-CRF model training procedure}

\label{alg:one}

\end{algorithm}