Character Identification on Multiparty Dialogue based on End-to-End Neural Coreference Resolution

Pu-Chin Chen {puchinchen@ucla.edu}
Aoxuan Li {douglasli@cs.ucla.edu}
Yutian Zhang {yutianzh0527@gmail.com}
Xin Liu {xinliu627@ucla.edu}

Abstract

Coreference resolution and entity linking are two important research topics in NLP, gaining growing attentions over years. In this project, we combined these two tasks together to accomplish a more complicated task, which is identifying the characters in multiparty dialogue. Different from previous traditional methods of coreference resolution, a state-of-the-art end-to-end model is adopted, which significantly outperforms other previous work. In addition, our entity-linking model is based on CNN instead of knowledge databases. Finally, we evaluated our models using the scripts from real TV shows. The results of coreference resolution are satisfactory, but the performance of entity linking does not live up to our expectations. We analyzed the possible underlying reasons, which pointing out the research direction in future work.

1 Introduction

Our main goal is to accomplish a shared task in SemEval 2018 - Task 4: Character Identification on Multiparty Dialogues. This tasks requires us to build a system which can identify different mentions in multiparty dialogues as corresponding characters in the show. This task is rather challenging, for cross-document entity resolution is imperative for identifying such mentions as real characters.

Literally, the character identification problem is tackled as a coreference resolution task with a further step on entity linking. In terms of this task, the baseline model generates mentions from a coreference system, and then each coreference chain is linked to a specific character identity. Both parts are implemented with agglomerative convolutional neural network in previous system (?; ?).Alternatively, we intend to use Bidirectional-LSTM with attention mechanism to address the same problem, which proves to have a satisfying performance in many NLP tasks. In our project, we apply the end-to-end neural coreference resolution model introduced by Lee and He, etc.(?). After obtaining the predicted clusters of mentions by the end-to-end coreference system, we choose the same algorithm of entity-linking as the one Chen proposed in his paper (?).

2 Related Work

2.1 Coreference resolution

The area of coreference resolution has resorted to the techniques of machine learning for many years. Still, the learning problem is challenging and, until very recently, hand-engineered systems built on top of automatically produced parse trees (?) outperformed all learning approaches.(?) showed that highly lexical learning approaches reverse this trend, and more recent neural models(?; ?; ?) have achieved noticeable performance gains. However, all of the aforementioned models still use parsers for head features and include highly engineered mention proposal algorithms. Such pipelined systems suffer from two major drawbacks. First, parsing mistakes can introduce cascading errors. Second, many hand-engineered rules do not generalize ideally to new languages or domains.

2.2 Entity Linking

Entity linking has traditionally relied heavily on knowledge databases, most notably, Wikipedia, for entities (?; ?).Although we do not make use of knowledge bases, our task is closely aligned to entity linking. Recent advances in entity linking are also applicable to our task since (**?**) use CNN to capture semantic similarity between a mention and an entity by comparing context of the mention with the description of the entity. This work justifies our usage of deep learning for character identification.

As indicated by the Dialogue State Tracking Challenges hosted by Microsoft (**?**), the dialogue tracking is an expanding task. The fact that an ongoing conversation can be dynamically tracked (**?**)is exciting and applicable to our task because the state of a conversation may yield significant hints for entity linking and coreference resolution. Speaker identification, a task similar to character identification, has already shown some success with partial dialogue tracking by dynamically identifying speakers at each turn in a dialogue using conditional random field models.

3 End-to-End Coreference Resolution

3.1 Basic Idea

Traditional coreference models usually rely on syntactic parsers from the perspective of the linguistics. In contrast, the end-to-end coreference resolution directly considers all the spans up to a maximum length in the document as potential mentions, computes the probability of possible ancestors (previous span) for each span, and directly optimizes the marginal likelihood of antecedent spans from gold coreference clusters.

Since the number of potential mentions is very large, it is impractical to score all span pairs. Therefore, this model uses unary mention scores to prune the space of pairs of spans(spans and antecedents), which significantly reduce the pairwise computations.

3.2 Model

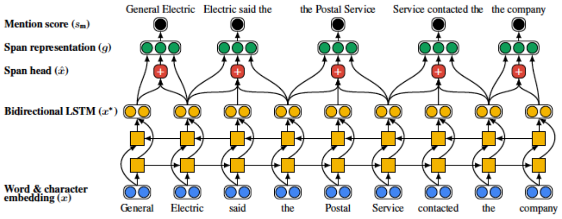

Figure 1: First step of end-to-end coreference resolution model

The first step is to compute embedding representations of spans for scoring potential entity mentions. A bidirectional LSTM is applied to encode the lexical information of both the inside and outside of each span. The model also includes a head-finding attention mechanism over words in each span to model head words. We use score to prune the spans, leaving only a manageable number of high-score spans for further steps.

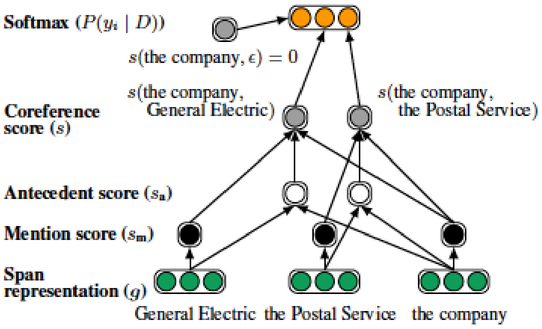
Softmax ($P(y_i \mid D)$)
s(the company, ϵ) = 0
s(the company, General Electric)
s(the company, the Postal Service)
Coreference score (s)
Antecedent score (s_a)
Mention score (s_m)
Span representation (g)
General Electric
the Postal Service
the company

Figure 2: Second step of the model

Second step is to compute antecedent scores from pairs of span representations. And then sum the mention scores of both spans and their pairwise antecedent score as the final coreference score of this pair of spans. The antecedent scoring function includes explicit element-wise similarity of each span pair and a learned 20-dimensional feature vector which encoding all features like speaker genre, span distance, mention width.

3.3 Learning

During the learning period, we optimize the marginal log-likelihood of all correct antecedents implied by the gold clustering:

$$log \prod_{i=1}^{N} \sum_{\hat{y} \in Y(i) \cap \text{GOLD}(i)} P(\hat{y}) \qquad (1)$$

where GOLD(i) is the set of spans in the gold cluster containing span i. If span i does not belong to a gold cluster or all gold antecedents have been pruned, GOLD(i) = $\{\varepsilon\}$.

4 Entity Linking

4.1 Introduction

Once the coreference resolution system is built and trained, we can predict co-references represented by clusters for each scene document. The next step should be making predictions from clusters to TV show character id. We model this process as an entity linking task. This section describes our entity linking model that takes the mention embeddings and the mention-pair embeddings generated by ACNN and classifies each mention to one of the character labels.

4.2 Model

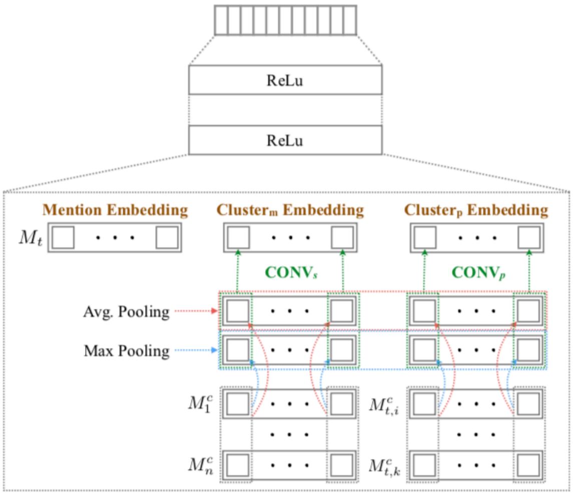

Figure 3: Character identification model

We prepare three embeddings generated by mentions which are predicted by previous coreference system. The first embedding is mention embedding; the second is embedding of the cluster including the mention; the third is generated by mention-pair embedding, which pairs the mention with reaming mention in the same cluster. Here are the formal formula:

$$\mathbf{R}_s(C_m) = [\mathbf{r}_s(m_1), \mathbf{r}_s(m_1), ..., \mathbf{r}_s(m_{|C_m|})] \quad (2)$$

$$\mathbf{R}_p(C_m, m) = [\mathbf{r}_p(m_i, m) | m_i \neq m] \quad (3)$$

In order to fix the input tensor size of both cluster embedding and mention-pair embedding, we perform avg-pooling and max-pooling for both embeddings. Then each of pooling layers is passed to a convolutional layer. Finally, we concatenate the mention embedding, cluster CNN embedding and mention-pair CNN embedding.

$$\mathbf{r}_s(C_m) = \text{CONV}_s(\begin{bmatrix} \text{avg\_pool}(\mathbf{R}_s(C_m)) \\ \text{max\_pool}(\mathbf{R}_s(C_m)) \end{bmatrix} \quad (4)$$

$$\mathbf{r}_p(C_m, m) = \text{CONV}_p(\begin{bmatrix} \text{avg\_pool}(\mathbf{R}_s(C_m, m)) \\ \text{max\_pool}(\mathbf{R}_s(C_m, m)) \end{bmatrix} \quad (5)$$

After concatenation, we feed them into a feed-forward neural network with two hidden layers. Final output is a softmax layer with the idmention of entity id number in Friends TV show.

4.3 Implementation

Our training data has 374 documents, which contains totally 13280 gold mentions that have gold entity label. Firstly, we need to get gold clusters and generate three embeddings, i.e. mention, cluster and mention pair embeddings. Since all gold clusters are labeled with entity ids, we can flat clusters to mentions while computing three embeddings. We can easily get mention embeddings from the pretrained coreference model.

5 Dataset and Evaluation Metrics

5.1 Data Description

The data comes from the scripts of the first two seasons of TV show Friends, which are already annotated for this task. Specifically, each season consists of episodes, and each episode is comprised of scenes.Furthermore, each scene is segmented into sentences.All datasets follow the CoNLL 2012 Shared Task data format. Each sentence is delimited by a new line and each column indicates the information regarding the word or the punctuation.

5.2 Data Split

Considering the amount of total data is not so sufficient, we simply split them into two datasets for different purposes, training and evaluation sets. The exact raito of our training set to our evaluation set is 4:1. The evaluation set contains seventy-four scenes in *Friends*.

5.3 Coreference Evaluation Metrics

The coreference results are evaluated with three metrics apropos coreference resolution:MUC, B^3,and $CEAF_e$.The precision, recall and F1 score of each metric approach are calculate separately and then the averages of them are obtained, which are utilized to compare against each other.

MUC
MUC(Vilain et al.,1995) concerns the number of pairwise links needed to be inserted or removed to map system responses to gold keys. The number of links shared by system and gold are calculated. In addition, minimum numbers of links needed to

	No Features			No Heads			Normal		
	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
MUC	79.14%	76.98%	81.42%	80.90%	80.38%	81.43%	80.84%	81.64%	80.71%
B^3	60.68%	55.22%	67.34%	64.38%	62.54%	66.34%	65.39%	68.89%	62.23%
$CEAF_e$	48.47%	48.21%	48.43%	52.89%	50.54%	55.48%	51.80%	53.51%	50.20%
AVG	62.76%	60.14%	65.73%	66.06%	64.49%	67.75%	66.02%	68.02%	64.17%

Table 1: Coreference Resolution Performances of No Features, No Heads, and Normal

describe coreference chains of the system and gold are computed as well.

B^3

Rather than evaluating coreference chains merely on their links,B^3(Bagga and Baldwin,1998) metric computes precision and recall on a mention level.System performance is evaluated by the average of all mentions scores.

$CEAF_e$

$CEAF_e$(Luo,2005) metric further clarifies the downside of B^3, in which entites can be used more than once during evaluation. Consequently, both multiple coreference chains of the same entity and chains with mentions of multiple entities are not penalized. To mitigate the aforementioned problem, $CEAF_e$ evaluates exclusively on the best one-to-one mapping between the systems and golds entities.

6 Results and Analysis

6.1 Ablation

To verify whether each component in our model is actually indispensable, we ablate various parts of the architecture and report the average F1 values on these experiments. We conducted one complete experiment and two ablation experiments, no heads and no features. In our dataset, the width of mention is one or two most of the time owing to the nature of daily dialogue, so the head-finding attention mechanism for each span seems to be useless. We select no head ablation to verify this guess. Besides, we also intent to demonstrate the significance of feature encoding.

6.2 Coreference Resolution Results

We conducted one complete experiment and two ablation experiments, no heads and no features. The performances of coreference resolution using these three models are shown in Table 1.

As we expected, there is almost no difference between the F1 value of normal model and F1 value of no heads model. What's more, the results unveil that the feature encoding indeed contributes to the better performance of coreference system, for the F1 value decreases conspicuously after the complete system is deprived of it.

The overall performance of coreference resolution is satisfactory and encouraging according to the F1 evaluation metric, which confirms the efficacy of the end-to-end system.

6.3 Entity Linking Results

We only got about 20% accuracy on character identification prediction, addressing that simply using combination of mention embeddings generated from coreference system is weak for predicting entity ids. It fails to contain cluster information even though the coreference system does be trained to optimize the clustering accuracy.

Another issue is that our models actually overfit. In the beginning, we used the ACNN model and overfit quite easily. Then we tried to reduce model complexities by only using one layer feedforward neural network on mention embeddings, getting rid of cluster embedding and mention pair embedding. However, the prediction precision is still low. This problem is arisen from two reasons. Firstly, this dataset is too small to use very complex model like deep neural networks. Secondly, consider our model, mention embeddings are extracted from merely word embeddings, which are insufficient to capture entity information.

7 Conclusion and Future work

In this work, we implemented a character identification system by integrating the neural coreference system with a CNN-based entity linking model. There are many open spaces for improvements. We should conduct the following experiments in the future:

Fortified Mention Embedding The coreference model, proposed by Lee and He, does not add speaker ids and other features until calculating antecedent scores. We can add these features earlier in mention embeddings after LSTM so as to directly utilize the information regarding speakers.

End-to-End Training Another improvement should be adding one layer for output entity ids after the coreference scores. The new loss function will be multiclass cross entropy. We can pre-train the coreference system and then train the entity linking part, by fixing the coreference model and optimize the loss. Inspired by (Clark and Manning, 2016), the further step will be designing loss function proposed in their work and boost the reliability of our system using reinforcement learning.