Robust Coreference Resolution and Entity Linking on Dialogues: Character Identification on TV Show Transcripts

Henry Y. Chen, Ethan Zhou, Jinho D. Choi
Math and Computer Science
Emory University
Atlanta, GA 30322, USA
{henry.chen, ethan.zhou, jinho.choi}@emory.edu

Abstract

This paper presents a novel approach to character identification, that is an entity linking task that maps mentions to characters in dialogues from TV show transcripts. We first augment and correct several cases of annotation errors in an existing corpus so the corpus is clearer and cleaner for statistical learning. We also introduce the agglomerative convolutional neural network that takes groups of features and learns mention and mention-pair embeddings for coreference resolution. We then propose another neural model that employs the embeddings learned and creates cluster embeddings for entity linking. Our coreference resolution model shows comparable results to other state-of-the-art systems. Our entity linking model significantly outperforms the previous work, showing the F1 score of 86.76% and the accuracy of 95.30% for character identification.

1 Introduction

Character identification (Chen and Choi, 2016) is a task that identifies each mention as a character in a multiparty dialogue.[1] Let a mention be a nominal referring to a human (e.g., *she*, *mom*, *Judy*), and an entity be a character in the dialogue. The objective is to assign each mention to an entity, who may or may not appear as a speaker in the dialogue. For the example in Table 1, the mention *comedian* is not one of the speakers in the dialogue; nonetheless, it clearly refers to a real person that may appear in some other dialogues. Identifying such mentions as actual characters requires cross-document entity resolution, which makes this task challenging.

Character identification can be viewed as a task of entity linking. Most of the previous work on entity linking focuses on Wikification (Mihalcea and Csomai, 2007a; Ratinov et al., 2011a; Guo et al., 2013). Unlike Wikification, entities in this task have no precompiled information from a knowledge base, which is another challenging aspect. This task is similar to coreference resolution in the sense that it groups mentions into entities, but distinct because it requires the identification of mention groups as real entities. Furthermore, even if it can be tackled as a coreference resolution task, only a few coreference resolution systems are designed to handle dialogues well (Rocha, 1999; Niraula et al., 2014) although several state-of-the-art systems have been proposed for the general domain (Peng et al., 2015; Clark and Manning, 2016; Wiseman et al., 2016).

Due to the nature of multiparty dialogues where speakers take turns to complete a context, character identification becomes a critical step to adapt higher-level NLP tasks (e.g., question answering, summarization) to this domain. This task can also bring another level of sophistication to intelligent personal assistants and intelligent tutoring systems. Perhaps the most challenging aspect comes from colloquial writing that consists of ironies, metaphors, or rhetorical questions. Despite all the challenges, we believe that the output of this task will enhance inference on dialogue contexts by providing finer-grained information about individuals.

In this paper, we augment and correct the existing corpus for character identification, and propose an end-to-end deep-learning system that combines neural models for coreference resolution and entity linking to tackle the task of character identification. The updated corpus and the source code of our models are published and publicly available.[2] This combined system utilizes the strengths from both

[1] The dialogues are extracted from TV show transcripts by the previous work (Chen and Choi, 2016).

[2] nlp.mathcs.emory.edu/character-mining/

Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 216–225,
Vancouver, Canada, August 3 - August 4, 2017.

Speaker	Utterance
Joey	Yeah, right! ... You$_1$ serious?
Rachel	Everything you$_2$ need to know is in that first kiss.
Chandler	Yeah. For us$_3$, it's like the stand-up comedian$_4$ you$_5$ have to sit through before the main dude$_6$ starts.
Ross	It's not that we$_7$ don't like the comedian$_8$, it's that ... that's not why we$_9$ bought the ticket.
$\{$You$_1\}$ → *Rachel*, $\{$us$_3$, we$_{7,9}\}$ → *Collective*, $\{$you$_{2,5}\}$ → *General*, $\{$comedian$_{4,8}\}$ → *Generic*, $\{$dude$_6\}$ → *Other*	

Table 1: An example of a multiparty dialogue extracted from the corpus.

models. We introduce a novel approach, agglomerative convolution neural network, for coreference resolution to learn mention, mention-pair, and cluster embeddings, and the results are taken as input to our entity linking model that assigns mentions to their real entities. Entities, including main characters and recurring support characters, are selected from a TV show to mimic a realistic scenario. To the best of our knowledge, this is the first end-to-end model that performs character identification on multiparty dialogues.

2 Related Work

The latest coreference systems employ advanced context features in tandem with deep networks to achieve state-of-the-art performance (Clark and Manning, 2016; Wiseman et al., 2015). Since our task is similar to coreference resolution, we take a similar approach to feature engineering by building mention and cluster embeddings with word embeddings (Clark and Manning, 2016) and include additional mention features described by Wiseman et al. (2015). We are motivated to use convolutional networks through the work of Wu and Ma (2017), but we distinguish our approach by using deep convolution to build embeddings for character identification.

Entity linking has traditionally relied heavily on knowledge databases, most notably, Wikipedia, for entities (Mihalcea and Csomai, 2007b; Ratinov et al., 2011b; Gattani et al., 2013; Francis-Landau et al., 2016).[3] 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 we see Francis-Landau et al. (2016) use convolutional nets to capture semantic similarity between a mention and an entity by comparing context of the mention with the description of the entity. This work validates our usage of deep learning for character identification.

Dialogue tracking has been an expanding task as shown by the Dialogue State Tracking Challenges hosted by Microsoft (Kim et al., 2015). That an ongoing conversation can be dynamically tracked (Henderson et al., 2013) 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 Corpus

The character identification corpus created by Chen and Choi (2016) includes entity annotation of personal mentions specific to the domain of multiparty dialogues. While the corpus covers a large amount of entities that appear in the first two seasons of the TV show, *Friends*, some of its annotation remains ambiguous, particularly around the label *Unknown*.

An example of *Unknown* mentions in a snippet of a conversation is provided in Table 1. Mentions *comedian*$_{4,8}$ and *dude*$_6$ are originally labeled *Unknown*, but they are two different entities such that their labels should be distinguished. Even though their entities are not immediately identifiable, the *Unknown* label provides no clarity; thus, mentions under this label needs to be subcategorized. We propose to disambiguate these *Unknown* mentions (Section 3.2), comprising 10% of the annotation. Such disambiguation allows finer-grained categories of entity annotations of mentions. We believe the resultant annotations are more realistic and can be used to train more robust model on character identification.

3.1 Corpus Correction

Before disambiguating the corpus, we find some recurring data malformations and errors in mention detection within the corpus. For example:

[3] This task is known as 'Wikification'.

Rachel: (To *guy* with a phone) Hello, excuse me.

The underlined action note is accidentally included in the utterance as a part of the dialogue due to a missing parentheses, and the mention *guy* is consequently incorporated into the corpus. These malformations are fixed, and mentions included are removed from the corpus manually before disambiguation. The correction is necessary since the inclusion of action notes is inconsistent throughout the corpus, and they are removed to avoid confusion for our models.

3.2 Corpus Disambiguation

Three labels are introduced to disambiguate *Unknown* mentions: *General*, *Generic*, and *Other*. *Generic* provides abstract groupings for unidentifiable entities, and each group is assigned a unique number for differentiation:

- *General*: Mention used in reference to a general case (e.g., *you*$_{2,5}$ in Table 1).
- *Generic*: Mention referring to a unidentifiable entity (e.g., *comedian*$_{4,8}$ in Table 1).
- *Other*: Mention referred to insignificant singleton entity (e.g., *dude*$_6$ in Table 1).

We perform this disambiguation manually with two main guidelines: only mentions originally labeled *Unknown* are included, and the labels introduced above are provided to annotators in addition to the known entities. We limit the *Generic* mention groups to 5 per iteration of disambiguation for simplicity, and the scenes that requires more than 5 groups are recursively annotated until all unknowns are disambiguated. Unlike the previous work, our annotators are familiar with the TV show, and the task takes about 3 weeks to complete.

	P	**S**	**C**	**G**	**N**	**O**	**Σ**
F1	5,101	2,610	1,259	109	152	184	9,306
F2	5,312	2,432	1,280	42	111	167	9,304
Σ	10,413	5,042	2,388	151	263	351	18,608

Table 2: Counts of disambiguated mentions. P/S: main and secondary character entities. C/G/N/O: *Collective*/*General*/*Generic*/*Other*.

4 Coreference Resolution

The task of character identification needs rich features extracted from mention clusters generated by a coreference resolution system. Thus, the end result of this task largely depends on the quality of the coreference resolution model. Several coreference resolution systems have been proposed and shown state-of-the-art performance (Pradhan et al., 2012); however, they are not necessarily designed for the genre of multiparty dialogue, where each document comprises utterances from multiple speakers.

This section describes a novel approach to coreference resolution using Convolutional Neural Networks (CNN). Our model takes groups of features incorporating several dialogue aspects, feeds them into deep convolution layers, and dynamically generates mention embeddings and mention-pair embeddings, which are used to create the cluster embeddings that significantly improve the performance of our entity linking model (Section 5).

4.1 Agglomerative CNN

Our coreference resolution model, Agglomerative Convolutional Neural Network (ACNN), takes advantage of deep layers in CNN. The model is called *agglomerative* since it aggregates multiple feature groups into several convolution layers for the generation of mention and mention-pair embeddings. Each layer aims to consolidate and learn different combinations of the input features, and additional features are included at each layer. The unique nature of our model allows incremental feature aggregations to create more robust embeddings. Figure 1 illustrates the complete architecture of ACNN.

The first part of the network learns the mention embedding for each of two mentions compared for a coreferent relation. Given two feature maps $\phi_e^k(m)$ and $\phi_d(m)$ where m is a mention, $\phi_e^k(m)$ extracts the embedding features based on word embeddings, and $\phi_d(m)$ extracts the discrete features (Table 3). The first convolution layer $\texttt{CONV}_1^k$ with n-gram filters of size d is applied to each embedding group k, and the result from each filter is max-pooled to generate a feature vector $\in \mathcal{R}^{1\times d}$. The second convolution layer $\texttt{CONV}_2$ is then applied to the 3D feature matrix $\in \mathcal{R}^{n\times d\times k}$ from the previous convolution layer on all embedding groups. The result of $\texttt{CONV}_2$ is max-pooled and concatenated with discrete features extracted by $\phi_d(m)$ to form the mention embedding $\mathbf{r}_s(m)$, defined as follows:

$$\mathbf{r}_s(m) = \texttt{CONV}_2(\begin{bmatrix}\texttt{CONV}_1^1(\phi_e^1(m))\\ \vdots \\ \texttt{CONV}_1^k(\phi_e^k(m))\end{bmatrix}) \parallel \phi_d(m)$$

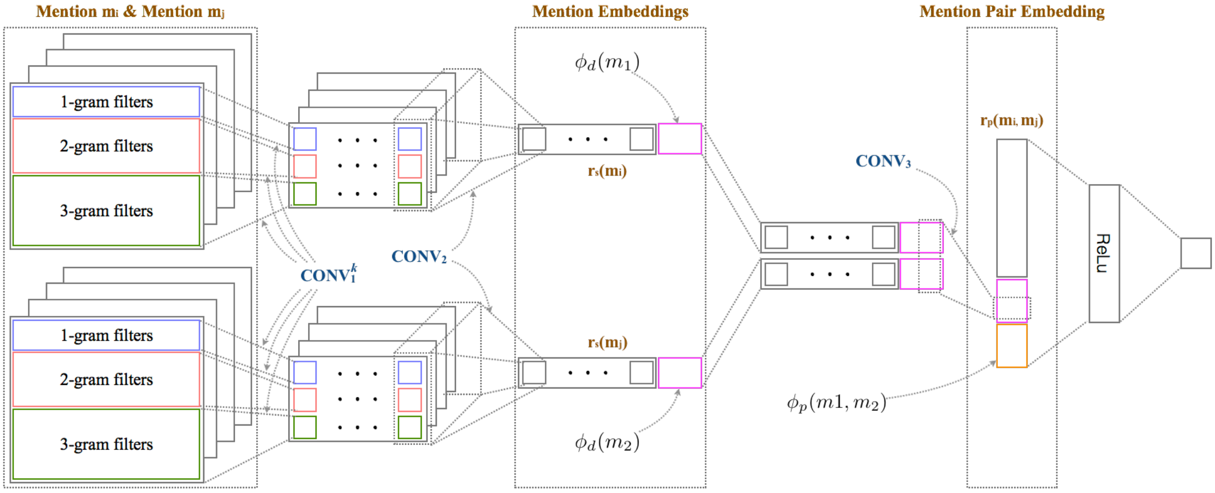

Figure 1: The overview of our agglomerative convolutional neural network.

The second part of the network utilizes the learned mention embedding $\mathbf{r}_s(m)$ to create the mention-pair embedding. Another feature map $\phi_p(m_i, m_j)$ is defined to extract pairwise features between mentions m_i and m_j (Table 3). The third convolution layer $\texttt{CONV}_3$ is applied to the stacked mention embeddings, $\mathbf{r}_s(m_i)$ and $\mathbf{r}_s(m_j)$. The result is max-pooled and concatenated with the pairwise features extracted by $\phi_p(m_i, m_j)$ to form the mention-pair embedding $\mathbf{r}_p(m_i, m_j)$, defined as follows:

$$\mathbf{r}_p(m_i, m_j) = \texttt{CONV}_3(\begin{bmatrix}\mathbf{r}_s(m_i)\\ \mathbf{r}_s(m_j)\end{bmatrix}) \parallel \phi_p(m_i, m_j)$$

The learned mention-pair embedding is put through the hidden layer with the linear rectifier activation function ($\texttt{ReLu}$) before applying the sigmoid function $\boldsymbol{\sigma}(m_i, m_j)$ to determine the coreferent relation between mentions m_i and m_j, defined as follows:

$$\mathbf{h}(x) = \texttt{ReLU}(\mathrm{w}_h \cdot x + b_h)$$
$$\boldsymbol{\sigma}(m_i, m_j) = \mathrm{sigmoid}(w_s \cdot \mathbf{h}(\mathbf{r}_p(m_i, m_j)) + b_s)$$

The purpose of the sigmoid function $\boldsymbol{\sigma}(m_i, m_j)$ is twofold. For each mention m_i, it performs binary classifications between m_i and m_j where $j \in [1, i)$. If $\max(\boldsymbol{\sigma}(m_i, m_j)) < 0.5$, the model considers no coreferent relation between m_i and any mention prior to it, and create a new cluster containing only m_i s.t. m_i becomes a singleton for the moment. If $\max(\boldsymbol{\sigma}(m_i, m_j)) \geq 0.5$, m_i is put to the existing cluster $\mathcal{C}_{m_k}$ that m_k belongs to, where m_k is $\arg_j \max(\boldsymbol{\sigma}(m_i, m_j))$. This formalism of mention clustering is defined as follows:

- If $\forall_{1 \leq j < i}. \max(\boldsymbol{\sigma}(m_i, m_j)) < 0.5$, then create a new cluster $\mathcal{C}_{m_i}$.
- If $\exists_{1 \leq j < i}. \max(\boldsymbol{\sigma}(m_i, m_j)) \geq 0.5$, then $\mathcal{C}_{m_k} \leftarrow \mathcal{C}_{m_k} \cup \{m_i\}$, where $m_k = \arg_j \max(\boldsymbol{\sigma}(m_i, m_j))$.

Table 3 shows feature templates used for our ACNN model. Sentence and utterance embeddings are the average vectors of all word embeddings in the sentence and utterance, respectively. Speaker embeddings are randomly generated using the Gaussian distribution. Gender and plurality information are from Bergsma and Lin (2006), and word animacy is from Durrett and Klein (2013).

Map	Features
$\phi_e^1(m)$	Embeddings of 1st three words in m
$\phi_e^2(m)$	Embeddings of 3 proceeding words of m Embeddings of 3 succeeding words of m Average embedding of all words in m
$\phi_e^3(m)$	Embeddings of 3 proceeding sentences Embeddings of 1 succeeding sentence Embedding of the current sentence
$\phi_e^4(m)$	Embeddings of 3 proceeding utterances Embeddings of 1 succeeding utterances Embeddings of the current utterance
$\phi_d(m)$	Avg. gender info. of all words in m Avg. plurality info. of all words in m Avg. word animacy of all words in m Embedding of the current speaker Embeddings of the previous 2 speakers
$\phi_p(m_i, m_j)$	Exact string match between m_i and m_j Relaxed string match between m_i and m_j Speaker match between m_i and m_j Mention distance between m_i and m_j Sentence distance between m_i and m_j

Table 3: Complete feature templates for ACNN. $\phi_e^k(m)$: embedding features, $\phi_d(m)$: discrete features, $\phi_p(m_i, m_j)$: pairwise features.

4.2 Configuration

For our experiments, word embeddings of dimension 50 are trained with FastText (Bojanowski et al., 2016) on the aggregation of New York Times,[4] Wikipedia,[5] and Amazon reviews.[6] The `tanh` activation function and a filter size of 280 is used for all convolution layers. A dropout rate of 0.8 is applied to all max-pooled convoluted results, and ℓ_2 regularization is applied to the sigmoid function. The hidden layer has the same dimension as the filter size. Binary labels of 0 and 1 are assigned to each mention-to-mention pair based on the gold cluster information. The model is trained on a mean squared error loss function with the RMSprop optimizer.

5 Entity Linking

Coreference resolution groups mentions into clusters; however, it does not assign character labels to the clusters, which is required for character identification. This section describes our entity linking model that takes the mention embeddings and the mention-pair embeddings generated ACNN and classifies each mention to one of the character labels (Figure 3). These embeddings are used to create cluster and cluster-mention embeddings through pooling, which give a significant improvement to character identification when included as features in our linker (Section 6).

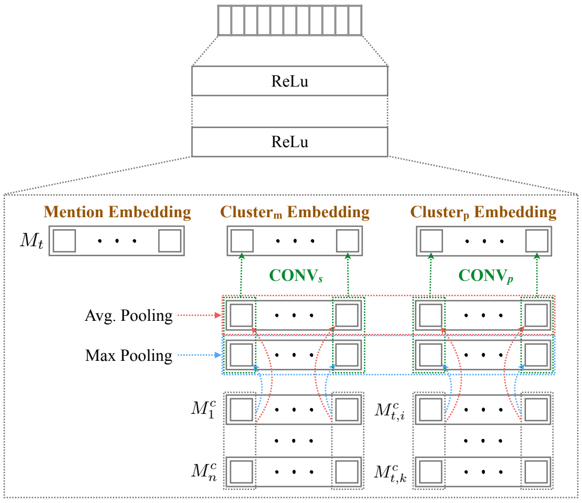

Figure 2: The overview of our entity linking model. Cluster$_\text{m}$ and Cluster$_\text{p}$ embeddings are derived from mention and mention-pair embeddings, resp.

Figure 2 illustrates our entity linking model based on a feed-forward neural network with two hidden layers. For each mention m, the model takes the mention embedding $\mathbf{r}_s(m)$ and two cluster embeddings derived from mention embeddings and mention-pair embeddings within the cluster $\mathcal{C}(m)$ (Section 5.2) and classifies m into one of the entity labels using the Softmax regression.

5.1 Cluster Embedding

Two types of cluster embeddings are derived to capture cluster information. Given a mention m and its cluster $\mathcal{C}_m$, cluster embedding $\mathbf{R}_s(\mathcal{C}_m)$ represents the collective mention embedding of all mentions within $\mathcal{C}_m$, and mention-cluster embedding $\mathbf{R}_p(\mathcal{C}_m, m)$ represents the collective mention-pair embedding between m and all the other mentions in $\mathcal{C}_m$ that are compared to m during coreference resolution ($\forall_i . \, m_i \in \mathcal{C}_m$):

$$\mathbf{R}_s(\mathcal{C}_m) = [\mathbf{r}_s(m_1), \mathbf{r}_s(m_2), ..., \ \mathbf{r}_s(m_{|\mathcal{C}_m|})]$$
$$\mathbf{R}_p(\mathcal{C}_m, m) = [\mathbf{r}_p(m_i, m) \mid m_i \neq m]$$

$\texttt{CONV}_s$ and $\texttt{CONV}_p$ are two separate convolution layers with unigram filters using the `tanh` activation. The results from these layers are max-pooled. The cluster embedding $\mathrm{r}_s(\mathcal{C}_m)$ and the mention-cluster embedding $\mathrm{r}_p(\mathcal{C}_m, m)$ are defined as follows:

$$\mathbf{r}_s(\mathcal{C}_m) = \texttt{CONV}_s(\begin{bmatrix} \texttt{avg\_pool}(\mathbf{R}_s(\mathcal{C}_m)) \\ \texttt{max\_pool}(\mathbf{R}_s(\mathcal{C}_m)) \end{bmatrix})$$
$$\mathbf{r}_p(\mathcal{C}_m, m) = \texttt{CONV}_p(\begin{bmatrix} \texttt{avg\_pool}(\mathbf{R}_p(\mathcal{C}_m, m)) \\ \texttt{max\_pool}(\mathbf{R}_p(\mathcal{C}_m, m)) \end{bmatrix})$$

The mention embedding, the cluster embedding, and the mention-cluster embedding are concatenated and fed into the network as input, and the scores of all character labels are activated as output.

5.2 Configuration

A dropout layer of rate 0.8 is applied to all inputs. The model is trained as a multi-class classifier with the categorical cross-entropy loss function and the RMSprop optimizer. All hidden layers use the `ReLU` activation function and have the same number of hidden units as the dimension of the mention embeddings. The convolution layers use the same filter sizes as the dimensions of input embeddings.

6 Experiments

Following Chen and Choi (2016), experiments are conducted on two tasks, coreference resolution and

[4] `catalog.ldc.upenn.edu/ldc2008t19`
[5] `dumps.wikimedia.org/enwiki/`
[6] `snap.stanford.edu/data/web-Amazon.html`

Model	Episode-Level					Scene-Level				
	MUC	B^3	$CEAF_e$	μ	$\lvert C \rvert$	MUC	B^3	$CEAF_e$	μ	$\lvert C \rvert$
Clark and Manning (2016)	89.58	69.12	**47.33**	**68.68**	15.19	**90.38**	76.79	56.95	74.70	8.13
Wiseman et al. (2016)	89.80	57.66	45.48	64.31	14.86	89.60	78.08	**65.95**	**77.88**	6.20
This work (ACNN)	**89.92**	**70.33**	44.09	68.11	16.40	88.09	**78.77**	59.72	75.53	7.49

Table 4: Coreference resolution results on the evaluation set (in %).
μ = (MUC + B^3 + $CEAF_e$) / 3. $|C|$: the average cluster size.

entity linking. Our coreference resolution model shows robust performance compared to other state-of-the-art systems (Section 6.2). Our entity linking model significantly outperforms the heuristic-based approach from the previous work (Section 6.3). All models are evaluated on the gold mentions to focus purely on the analysis of these two tasks.

6.1 Data Split

The corpus is split into the training, development, and evaluation sets (Table 5). For the episode-level, all mentions referring to the same character in each episode are grouped into one cluster (C_{Epi}). For the scene-level, this grouping is done by each scene such that there can be multiple mention clusters that refer to the same character within an episode (C_{Sce}). Ambiguous mention types such as *collective*, *general*, and *other* are excluded from our experiments (Section 3); including those mentions requires developing different resolution models that we shall explore in the future.

	E	S	DC	C_E	C_S	M
TRN	38	362	371	820	2,026	12,842
DEV	3	28	44	58	159	991
TST	5	58	80	113	301	1,885
Total	46	448	444	991	2,486	15,718

Table 5: The training (TRN), development (DEV), and evaluation (TST) sets. E/S/DC/C_E/C_S/M: the numbers of episodes, scenes, distinct characters, episode/scene-level clusters, and mentions.

For entity linking, entity labels are predetermined by collecting characters that appear in all three sets; characters that do not appear in any of the three sets are put together and labeled as *Unknown*. This is reasonable because it is not possible for a statistical model to learn about characters that do not appear in the training set. Likewise, characters that appear in the training set but not in the other sets cannot be developed or evaluated. A total of ten labels are used for entity linking that consist of the top-9 most frequently appeared characters across all sets and *unknown* (Figure 3).

6.2 Coreference Resolution

To benchmark the robustness of our ACNN model (Section 4), two state-of-the-art coreference resolution systems are also experimented. Episode and scene-level models are developed separately for all three systems using the same dataset in Table 5. All system outputs are evaluated with the MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_e$ (Luo, 2005) metrics suggested by the CoNLL'12 shared task (Pradhan et al., 2012). The average score of five trials is reported for each metric to minimize variance because these systems use neural network approaches with random initialization to produce varying results per trial (Table 4).

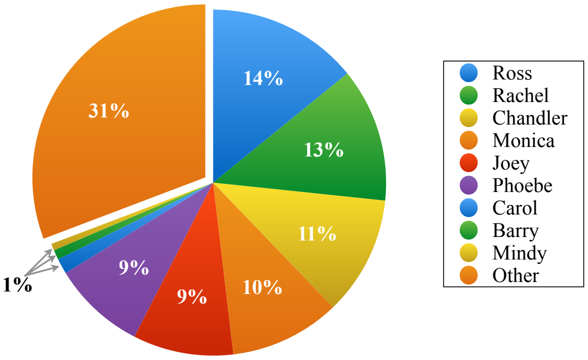

Figure 3: Character labels used for entity linking.

Comparison between the State-of-the-Art

When trained and evaluated on our dataset, both the Stanford (Clark and Manning, 2016) and the Harvard (Wiseman et al., 2016) systems give comparable results to their performance on the CoNLL'12 dataset.[7] The Stanford system using its pre-trained model gives the μ scores of 47.67% and 64.14% for the episode and scene-level respectively, which signifies the importance of the in-domain training data.

[7] The Stanford and the Harvard systems reported μ scores of 65.73% and 64.21% on the CoNLL'12 dataset, respectively.

Model		Ross	Joey	Chandler	Monica	Phoebe	Rachel	Carol	Mindy	Barry	Unk.	Avg	Acc
E	B	57.54	80.94	64.91	89.82	87.86	76.47	30.14	0	16.67	70.24	57.46	72.52
	ME	72.81	80.31	82.43	79.78	82.71	82.94	44.84	20.00	53.05	76.23	67.51	77.80
	CE	93.46	97.90	98.23	95.42	98.24	95.02	100.00	0	95.65	93.71	**86.76**	**95.30**
S	B	60.00	69.09	61.05	72.51	57.27	78.77	34.38	0	11.76	67.62	51.24	66.68
	ME	74.75	81.76	80.71	88.83	84.33	85.43	53.15	20.00	62.90	80.82	71.27	81.07
	CE	91.29	90.64	86.33	94.10	85.41	90.16	65.35	18.71	83.45	85.82	**79.12**	**87.64**

Table 6: Entity linking results on the evaluation set (in %). The F1 score is reported for each character. E/S: episode/scene level. Unk.: *unknown*. Avg: the macro-average F1 score between all characters. Acc: (the number of correctly labeled mentions) / (the total number of mentions).

All systems show higher scores for the scene-level than the episode-level consistently, which confirms the difficulty of this task on larger documents.

Although both systems take advantage of global cluster features, they reveal different strengths on resolving mentions with respect to the cluster size. The Stanford system excels for the episode-level, which is primarily attributed to the cluster-based nature of this system; it is able to find more accurate coreferent chains when the clusters are larger. The Harvard system performs best for the scene-level, indicating that its neural architecture with Long Short-Term Memory cells captures more meaningful cluster features when the clusters are smaller.

Comparison to Agglomerative CNN

In comparison to the other state-of-the-art systems, our ACNN model shows competitive performance; it gives the highest B^3 and comparable μ scores for both episode and scene levels. We measure the average cluster size produced by each system for further analysis (|C| in Table 4). The Harvard system produces smaller clusters than the other two systems. Such a tendency gives more pure clusters, favored by the CEAF_e metric for the scene-level. However, it is prone to breaking up too many links, which leads to poor performance in the B^3 evaluation on the episode-level.

The performance of our model is encouraging although coreference resolution is not the end goal. We design this model to automatically generate mention embeddings and mention-pair embeddings that are used to construct cluster features for entity linking. However, even though this model's success in coreference resolution is not our final objective, its success directly correlates to the success of entity linking because of the similarity between these two tasks. Due to the similar nature of these two tasks, the success of coreference resolution directly correlates to that of entity linking. These embeddings are the essence of our entity linking model, leading to a huge improvement.

6.3 Entity Linking

The heuristic-based approach proposed by Chen and Choi (2016) is adapted to establish the baseline. Two statistical models are experimented for both the episode and scene levels, one using only mention embeddings and the other using both mention embeddings and cluster embeddings (Section 5). All models are evaluated with the F1 scores of character labels, the macro-average F1 scores between all labels, and the label accuracies. The average scores of five trials are reported in Table 6.

B: Baseline Model

The heuristic-based approach is applied to the mention clusters found by our coreference resolution model. Two rules, $^{1)}$proper noun and $^{2)}$first-person pronoun matches, are used to assign character labels to all mentions. The label of each cluster is then determined by the majority vote between the mention labels within the cluster. Finally, the cluster label is assigned to all mentions in that cluster. This model performs better when it is applied to the episode-level clusters because larger clusters provide more mention labels, which makes the majority vote more reliable.

ME: Mention Embedding Model

This model takes advantage of the mention embeddings generated by our ACNN model. Compared to the baseline, it gives over a 21% higher average F1 score, and over a 15% higher label accuracy for the episode and the scene levels, respectively. Interestingly, this model shows higher performance for the scene-level, which is not the case for the other two models. This implies that the mention embeddings learned from scene-level documents are more informative than those learned from episode-level ones.

		System										
		Ross	**Joey**	**Chandler**	**Monica**	**Phoebe**	**Rachel**	**Carol**	**Mindy**	**Barry**	**Unk.**	**Σ**
Gold	**Ross**	182					7				1	190
	Joey		186								6	192
	Chandler			235								235
	Monica			1	200							201
	Phoebe					141					2	143
	Rachel				2		237					239
	Carol							49				49
	Mindy								0		9	9
	Barry									11		11
	Other	11	1	11	21	4		5		1	562	616
	Σ	193	187	247	223	145	244	54	0	12	580	1,885

Table 7: The confusion matrix between gold and system annotation for all character labels (in #).

This case is also reflected on its coreference resolution performance where the scene-level scores are higher than the episode-level scores (Table 4).

CE: Cluster Embedding Model

While the mention embeddings give a significant improvement over the baseline, further improvement is made when they are coupled with the cluster and mention-cluster embeddings. The episode-level cluster embedding model shows an average F1 score of 86.76% and a label accuracy of 95.30%, which is another 15% improvement, suggesting a practical use of this model in real applications. A couple of important observations are made:

- Cluster and mention-cluster embeddings, although learned during coreference resolution, are crucial for entity linking such that a coreference resolution model specifically designed for multiparty dialogues is necessary to build the state-of-the-art entity linking model for this genre.
- Clusters generated from the episode-level documents provide more information than those from the scene-level do, which aligns with the conclusion made by Chen and Choi (2016).

Error Analysis

An error analysis is performed on the episode-level cluster embedding model. From the confusion matrix in Table 7, two common system errors are detected. First, most of the mispredictions identify *Unknown* as specific characters. Second, the performance on the secondary characters, *Carol*, *Mindy*, and *Barry*, is subpar with respect to other entities. This subpar performance likely stems from a paucity of appearances by these secondary characters. For example, *Mindy* constitutes 1% of the dataset (Figure 3) and has only nine occurrences in the evaluation set. Our best model is robust in identifying the primary characters, showing an average F1 score of 96.38% and an accuracy of 98.42% on the evaluation set.

7 Conclusion

In this paper, we explore a relatively new task, character identification on multiparty dialogues, and introduce a novel perspective on approaching the task with coreference resolution and entity linking. We improve and augment finer-grained annotation over the existing corpus that simulates real conversations. We propose a deep convolutional neural network to agglomerate groups of features into mention, mention-pair, cluster, and mention-cluster embeddings that are optimized for entity prediction. Our coreference resolution result shows an improvement on the updated version of the corpus. Our entity linking result reaches to the accuracy that is sufficient for real-world applications.

To the best of our knowledge, our work is the first time that such deep convolution layers have been used for training mention and cluster embeddings. Our results show that the generation of these embeddings is crucial for the success of entity linking on multiparty dialogues. For future work, we will continue to increase the size of the corpus with high-quality and disambiguated annotation. We also wish to improve the embeddings to represent plural and collective mentions, thus we can build upon our entity linking model incorporating many-to-many linkings between entities and mentions.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*. Citeseer, volume 1, pages 563–566.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sydney, Australia, pages 33–40.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Yu-Hsin Chen and Jinho D. Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Los Angeles, pages 90–100. http://www.aclweb.org/anthology/W16-3612.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2256–2262. https://aclweb.org/anthology/D16-1245.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 1256–1261. http://www.aclweb.org/anthology/N16-1150.

Abhishek Gattani, Digvijay S. Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. 2013. Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proc. VLDB Endow.* 6(11):1126–1137. https://doi.org/10.14778/2536222.2536237.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. NAACL, pages 1020–1030.

Matthew Henderson, Blaise Thomson, and Steve Young. 2013. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*. Association for Computational Linguistics, Metz, France, pages 467–471. http://www.aclweb.org/anthology/W13-4073.

Seokhwan Kim, Luis Fernando D́Haro, Rafael E. Banchs, Jason D. Williams, and Matthew Henderson. 2015. The Fourth Dialog State Tracking Challenge. In *Proceedings of the 4th Dialog State Tracking Challenge*. DSTC4.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 25–32.

Rada Mihalcea and Andras Csomai. 2007a. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. CIKM'07, pages 233–242.

Rada Mihalcea and Andras Csomai. 2007b. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '07, pages 233–242. https://doi.org/10.1145/1321440.1321475.

Nobal B. Niraula, Vasile Rus, Rajendra Banjade, Dan Stefanescu, William Baggett, and Brent Morgan. 2014. The DARE Corpus: A Resource for Anaphora Resolution in Dialogue Based Intelligent Tutoring Systems. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*. LREC'14, pages 3199–3203.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A Joint Framework for Coreference Resolution and Mention Head Detection. In *Proceedings of the 9th Conference on Computational Natural Language Learning*. CoNLL'15, pages 12–21.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning: Shared Task*. CoNLL'12, pages 1–40.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011a. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. ACL'11, pages 1375–1384.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011b. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 1375–1384. http://dl.acm.org/citation.cfm?id=2002472.2002642.

Marco Rocha. 1999. Coreference Resolution in Dialogues in English and Portuguese. In *Proceedings of the Workshop on Coreference and Its Applications*. CorefApp'99, pages 53–60.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*. Association for Computational Linguistics, pages 45–52.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1416–1426. http://www.aclweb.org/anthology/P15-1137.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 994–1004. http://www.aclweb.org/anthology/N16-1114.

J. L. Wu and W. Y. Ma. 2017. A deep learning framework for coreference resolution based on convolutional neural network. In *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. pages 61–64. https://doi.org/10.1109/ICSC.2017.57.