They Exist! Introducing Plural Mentions to Coreference Resolution and Entity Linking

Ethan Zhou
Computer Science
Emory University
Atlanta, GA 30322
ethan.zhou@emory.edu

Jinho D. Choi
Computer Science
Emory University
Atlanta, GA 30322
jinho.choi@emory.edu

Abstract

This paper analyzes arguably the most challenging yet under-explored aspect of resolution tasks such as coreference resolution and entity linking, that is the resolution of plural mentions. Unlike singular mentions each of which represents one entity, plural mentions stand for multiple entities. To tackle this aspect, we take the character identification corpus from the SemEval 2018 shared task that consists of entity annotation for singular mentions, and expand it by adding annotation for plural mentions. We then introduce a novel coreference resolution algorithm that selectively creates clusters to handle both singular and plural mentions, and also a deep learning-based entity linking model that jointly handles both types of mentions through multi-task learning. Adjusted evaluation metrics are proposed for these tasks as well to handle the uniqueness of plural mentions. Our experiments show that the new coreference resolution and entity linking models significantly outperform traditional models designed only for singular mentions. To the best of our knowledge, this is the first time that plural mentions are thoroughly analyzed for these two resolution tasks.

1 Introduction

Resolution tasks such as coreference resolution and entity linking are challenging because they require a holistic view of a document (or across multiple documents) to find correct entities. Although many models have been proposed for these tasks (Clark and Manning, 2016; Francis-Landau et al., 2016; Wiseman et al., 2016; Gupta et al., 2017; Lee et al., 2017), most of them are focused on singular mentions such that they are insufficient for resolving the other type of mentions, plural, although the amount of plural mentions is not negligible in practice.[1] Table 1 illustrates how mentions are annotated for coreference resolution by the CoNLL'12 shared task (Pradhan et al., 2012) and our proposed work. In the CoNLL'12 annotation, the plural mention $They_8$ is grouped with the noun phrase [$Mary_1$ and $John_2$]$_3$; however, the other plural mention We_7 becomes a singleton because there is no noun phrase representing such an entity. Since CoNLL'12 limits each plural mention to be linked to a single noun phrase, it loses connections to individual entities that exist within the document but not grouped as a noun phrase.

Document	[$Mary_1$ and $John_2$]$_3$ came to see me_4 yesterday. She_5 looked happy, and so did he_6. We_7 had a great time together. $They_8$ left around noon.
CoNLL'12	{$Mary_1$, She_5}, {$John_2$, he_6}, {[$Mary_1$ and $John_2$]$_3$, $They_8$}, {me_4}, {We_7}
Our Work	{$Mary_1$, She_5, We_7, $They_8$}, {$John_2$, he_6, We_7, $They_8$}, {me_4, We_7}

Table 1: Snippets of how mentions are annotated by the CoNLL'12 shared task and our work.

In our work, the plural mentions We_7 and $They_8$ are linked to multiple entities that those mentions refer to. This allows higher-level NLP tasks such as question answering or machine translation to reason more explicitly about those entities while adding another level of challenges to the resolution tasks. In this paper, we first present the annotation scheme for resolving plural mentions that is used to expand the corpus

[1] A singular mention is a noun phrase that refers to exactly one entity while a plural mention is one that refers to multiple entities.

Proceedings of the 27th International Conference on Computational Linguistics, pages 24–34
Santa Fe, New Mexico, USA, August 20-26, 2018.

provided by the Character Mining project (Section 3). We then introduce a novel algorithm for coreference resolution that selectively creates clusters for singular and plural mentions, as well as evaluation metrics to handle plural mentions for coreference resolution (Section 4). We also present a new deep learning-based entity linking model that jointly identifies both singular and plural mentions (Section 5). All models are evaluated on our dataset (Section 6); the experiments reveal significant improvement from our new models compared to the previous state-of-the-art models dedicated for singular mentions. As far as we can tell, this is the first time that such annotation for plural mentions is provided in a large enough scale that deep learning models can be trained on, at the same time, machine learning models are developed to achieve promising results for the resolution of plural mentions.

2 Related Work

Chen and Choi (2016) were the first to introduce the task of character identification and provided a new corpus based on TV show transcripts. Given a dialogue transcribed in text where all mentions are detected, character identification aims to find the entity for each personal mention, who may or may not be active in the dialogue. Unlike most other entity linking tasks focusing on Wikification, this task is challenging because it is dialogue-based where the entities are general characters in the show. This corpus was later expanded by Chen et al. (2017) who added annotation for the ambiguous entity types. In this work, we expanded the corpus further by doubling the size of the annotation and adding new annotation for plurals.

The character identification corpus can be used for both coreference resolution and entity linking tasks. Our approach to coreference resolution was partially motivated by the previous works, Clark and Manning (2016) and Durrett et al. (2013), who tackled the general cases of coreference resolution including plurals; however, since their approaches were based on the annotation provided by CoNLL'12, they did not handle plural mentions to our satisfaction (Table 1). Jain et al. (2004) presented a rule-based system for resolving plural mentions, which was limited to unambiguous plural types. Our work is distinguished because we handle both ambiguous and unambiguous types of plural mentions, which makes it more challenging. Chen et al. (2017) presented an entity linking model that identified the real entity of each singular mention, which we adapted to develop a new multi-task learning model that jointly handles singulars and plurals.

3 Corpus

3.1 Annotation

The Character Mining project provides transcripts from the TV show *Friends* for all ten seasons in JSON.[2] A subset of the first two seasons of this show was annotated for the task of character identification by Chen et al. (2017), who made it publicly available through the International Workshop on Semantic Evaluation (SemEval 2018).[3] Given this annotation, we expanded the corpus as follows:

1. We realized that about 20% of the first two seasons were not covered by the previous annotation. Following the annotation guidelines suggested by Chen et al. (2017), we completed the annotation for the first two seasons and further annotated two more seasons. As a result, the first four seasons are completely annotated for character identification in our corpus.

2. There were quite a few mismatches among the speaker and the entity labels in the previous annotation. For instance, while mentions were annotated by the entity's full name such as `Monica_Geller`, some utterances were paired with speaker labels represented by only the first name, `Monica`, which could cause confusions for machine learning models. We manually went through the entire annotation and made sure the speaker and the entity labels were coherent across all seasons.

3. The previous annotation consisted of only singular mentions such that each mention was guaranteed to be linked to exactly one entity. We annotated plural mentions for the first four seasons through crowdsourcing. Unlike singular mentions that were automatically recognized by the heuristic-based

[2] Character Mining: `https://github.com/emorynlp/character-mining`

[3] SemEval 2018 Task 4: `https://competitions.codalab.org/competitions/17310`

mention detector (Chen and Choi, 2016), plural mentions in our corpus were manually detected by the crowd workers who were also asked to link each plural mention to a set of its referent entities.

The annotation guidelines used for singular mentions are adapted to annotate plural mentions as well such that the only difference in annotation between these two types of mentions is the number of entities to which the mentions refer. Formally, each mention m is annotated with a set of entities E, where each element in E belongs to one of the following four groups:

1. Known entities: include all the primary and secondary characters recurring in the show.
2. GENERIC: indicates actual characters in the show whose identities are unknown across the show: e.g., That *waitress* is really cute, I am going to ask *her* out.
3. GENERAL: indicates mentions referring to a general case rather than a specific entity: e.g., The ideal *guy* you look for doesn't exist.
4. OTHER: indicates actual characters in the show whose identities are unknown in this dialogue but revealed in some other dialogue.

The COLLECTIVE type, used to distinguish the plural usage of the pronoun *you* in the previous annotation, is discarded in our annotation because each *you* is now annotated with a set of entities such that the plural usage can be deterministically distinguished by the size of its entity set.

Speaker	**Utterance**
Jack	And *I*$_1$ read about these *women*$_2$ trying it all, and *I*$_3$ thank God '*Our*$_4$ *Harmonica*$_5$' doesn't have this problem.
Monica	So, *Ross*$_6$, what's going on with *you*$_7$ two? Any stories? No little anecdotes to share with *mom*$_8$ and *dad*$_9$?
Ross	Okay, *I*$_{10}$ just got this from the *guy*$_{11}$ next to *me*$_{12}$. *He*$_{13}$ was selling a whole bunch of stuff.

$\{I_1, I_3, Our_4, dad_9\} \rightarrow$ Jack $\{Our_4, mom_8\} \rightarrow$ Judy, $\{Harmonica_5\} \rightarrow$ Monica, $\{Ross_6, you_7, I_{10}, me_{12}\} \rightarrow$ Ross,
$\{women_2\} \rightarrow$ GENERAL, $\{you_7\} \rightarrow$ OTHER, $\{guy_{11}, He_{13}\} \rightarrow$ MAN_1

Table 2: An example of entity annotation in our corpus, where *Our*$_4$ and *you*$_7$ are the plural mentions.

Table 2 shows examples of all types of entities for both singular and plural mentions. The mention *women*$_2$ does not refer to any specific character so it is identified as GENERAL. Both the mentions *guy*$_{11}$ and *He*$_{13}$ refer to a specific person whose identity is never revealed so it is annotated with the generic type, MAN_1. There are two plural mentions, *Our*$_4$ and *you*$_7$, which are handled differently. All entities of *Our*$_4$ can be identified from the context of this dialogue so it is annotated with the known entities Jack and Judy. However, only one of *you*$_7$ can be identified in this context so it is annotated with the known entity Ross and also OTHER, implying that it refers to some other entity that can be identified in a separate dialogue. This method is used to distinguish non-immediately identifiable entities from the generic case of MAN_1 whose identity is unknown across the entire show.

3.2 Analytics

Table 3 shows the statistics of our corpus. Compared to the previous annotation including 18,608 mentions, our corpus is comprised of 47,367 annotated mentions, which is 2.5 times larger. Plural mentions together compose about 9% of the entire dataset, which is significant enough to make a difference in resolution. Each cluster contains about 6 mentions on average when each scene is treated as an independent dialogue.

All mentions were double-annotated by crowd workers. From this double-annotation, Cohen's kappa score of 56.88% was achieved for plural mentions, which was about 20% lower than the one achieved for singular mentions (Chen and Choi, 2016). The lower inter-annotator agreement was expected due to the high complexity of this task. A subset of the disagreed annotation was manually adjudicated by experts, from which we found that taking the union of the entity sets annotated by two workers would effectively give the correct set of entities for each of those disagreed plural mentions. Thus, a vast amount of plural mentions were pseudo-adjudicated by taking their unions of double-annotation.

Season	General				Mention			Entity	
	Episode	Scene	Utterance	Speaker	Singular	Plural	Total	Cluster	Type
1	24	326	5,968	107	10,313	1,147	11,460	2,162	270
2	24	293	5,747	107	10,521	1,156	11,677	1,934	285
3	25	348	6,495	108	11,458	907	12,365	1,925	230
4	24	334	6,318	100	10,726	1,139	11,865	1,881	175
Total	97	1,301	24,528	331	43,018	4,349	47,367	7,902	781

Table 3: The overall statistics of our corpus. All columns show raw counts except that the speaker column and the type column in the entity section give the set counts of all speakers and entities, respectively.

Table 4 shows the distributions of entity types. The primary characters compose about 67% of all mentions whereas the ambiguous types together compose about 8.6%, which implies that the majority of mentions can be linked to known entities. Notice that the total count of `GENERAL` increases by 554 from Seasons 1-2 to 3-4, whereas the total count of `OTHER` decreases by 654 for those seasons; these two ambiguous entity types are easily confused because they do not refer to any specific entity within the dialogue. Considering that annotation tasks for the first two seasons were mostly conducted by Chen et al. (2017) whereas the next two seasons were conducted by us, it is possible that our crowdsourcing instructions were more biased towards `GENERAL` than `OTHER`, which we will analyze in the future.

Season	Known Entities		Ambiguous Entities			Total
	Primary	Secondary	**GENERIC**	**GENERAL**	**OTHER**	
1	9,247	3,616	214	641	463	14,181
2	9,591	3,704	184	598	455	14,532
3	9,491	3,512	200	896	136	14,235
4	9,807	3,181	112	897	128	14,125
Total	38,136	14,013	710	3,032	1,182	57,073

Table 4: The distributions of entity types. Each column shows the number of mentions annotated with the corresponding entity type. Note that the total number of mentions here is different from the one in Table 3 (57,073 vs. 47,367) because each plural mention is counted more than once in this table.

4 Coreference Resolution

The presence of plural mentions brings up several challenges for coreference resolution. First, the search scope becomes broader. For each mention m_j, a typical coreference resolution system would find another mention m_i that is referent to m_j, and assigns m_j to the cluster C_i that m_i belongs to if it exists; otherwise, creates a new cluster and assigns both m_i and m_j to that cluster.[4] As soon as m_j is assigned, the search can stop for m_j. This strategy works for singular mentions but fails with plural mentions because they can be assigned to more than one cluster. Second, the referent relations are no longer transitive. Let $m_i \leftarrow m_j$, $m_i \rightarrow m_j$, $m_i \leftrightarrows m_j$ stand for referent relations such that m_j is referent to m_i, m_i is referent to m_j, m_i is coreferent to m_j, respectively. Then, $m_i \leftrightarrows m_j$ and $m_j \leftrightarrows m_k$ would imply $m_i \leftrightarrows m_k$ for singular mentions, but this transitivity fails with plural mentions when m_j belongs to two different clusters $C_i = \{m_i, m_j\}$ and $C_k = \{m_j, m_k\}$ such that m_i and m_k have no referent relation. Third, some of the popular evaluation metrics for coreference resolution such as B^3 (Bagga and Baldwin, 1998) are not necessarily designed for plural mentions such that they need to be revisited.

Section 4.1 introduces our new coreference resolution algorithm that selectively creates clusters with respect to different mention types. This algorithm ensures singular mentions representing different entities get assigned to separate clusters. For example, let m_p be a plural mention and m_i be a singular mention such that $m_p \rightarrow m_i$. When the referent relation is found, the cluster C_i is created and both m_p and m_i are assigned to C_i. Let m_j be another singular mention such that $m_p \rightarrow m_j$. Now, the algorithm must decide

[4] The term 'cluster‘ indicates a group of mentions that refer to the same entity within a document such that each cluster represents a distinct entity although a cluster in one document can represent the same entity as another cluster in a different document.

whether to assign m_j to C_i or create another cluster C_j for m_j. If $m_i \leftrightarrows m_j$, m_j should be assigned to C_i; otherwise to C_j. Our algorithm allows a model to learn this decision during training so that the clusters can be created accordingly during decoding. Section 4.3 describes how existing evaluation metrics can be adjusted to evaluate both singular and plural mentions for coreference resolution, which is the first time that these metrics are adapted for plural mentions linked to multiple entities.

4.1 Algorithm

For each mention m_j, our algorithm compares it against all of the preceding mentions m_i to determine whether or not they are referent, where i and j are the ordered indices such that $0 < i < j$. Additionally, two more mentions, m_g and m_o, are compared to m_j that represent the `GENERAL` and the `OTHER` types, respectively (Section 3.1). For each mention pair (m_i, m_j), the algorithm assigns one of the following three labels for multi-classification:

1. `N`: m_i is not referent to m_j.

2. `L`: m_j gets assigned to the cluster that m_i belongs to. If m_i does not yet belong to any cluster, a new cluster C_i is created and both m_i and m_j are assigned to C_i.

3. `R`: m_i gets assigned to the cluster that m_j belongs to. If m_j does not yet belong to any cluster, a new cluster C_j is created and both m_i and m_j are assigned to C_j.

During training, labels are determined by consulting the oracle. `L` is labeled if m_i is a singular mention. `R` is labeled if m_i is plural and m_j is singular. `N` is labeled for all the other cases. Notice that this algorithm does not allow any plural mention to be directly linked to another plural mention; in other words, it does not create any cluster consisting of only plural mentions. Plural mentions can still be indirectly linked through clusters created for singular mentions. The creation of clusters comprising only plural mentions would not help identifying the known entities of those mentions, which defeats the purpose of character identification. It is possible to link plural mentions directly by using the `GENERIC` type (Section 3.1), which is not adapted to annotate entities for plural mentions in the current annotation scheme.

$[\boldsymbol{m_i}] \rightarrow \{\texttt{N, L, R}\}$	$\boldsymbol{m_j}$	**Clusters**
$[\texttt{G}, \texttt{O}] \rightarrow \texttt{N}$	1	$\emptyset_g, \emptyset_o$
$[\texttt{O}, 1] \rightarrow \texttt{N}, [\texttt{G}] \rightarrow \texttt{L}$	2	$\{2\}_g, \emptyset_o$
$[\texttt{G}, \texttt{O}, 2] \rightarrow \texttt{N}, [1] \rightarrow \texttt{L}$	3	$\{2\}_g, \emptyset_o, \{1, 3\}_1$
$[\texttt{G}, \texttt{O}, 2] \rightarrow \texttt{N}, [1, 3] \rightarrow \texttt{L}$	4	$\{2\}_g, \emptyset_o, \{1, 3, 4\}_1$
$[\texttt{G}, \texttt{O}, 1..4] \rightarrow \texttt{N}$	5	$\{2\}_g, \emptyset_o, \{1, 3, 4\}_1$
$[\texttt{G}, \texttt{O}, 1..5] \rightarrow \texttt{N}$	6	$\{2\}_g, \emptyset_o, \{1, 3, 4\}_1$
$[\texttt{G}, 1..5] \rightarrow \texttt{N}, [\texttt{O}, 6] \rightarrow \texttt{L}$	7	$\{2\}_g, \{7\}_o, \{1, 3, 4\}_1, \{6, 7\}_6$
$[\texttt{G}, \texttt{O}, 1..3, 5..7] \rightarrow \texttt{N}, [4] \rightarrow \texttt{R}$	8	$\{2\}_g, \{7\}_o, \{1, 3, 4\}_1, \{6, 7\}_6, \{4, 8\}_8$
$[\texttt{G}, \texttt{O}, 2, 5..8] \rightarrow \texttt{N}, [1, 3, 4] \rightarrow \texttt{L}$	9	$\{2\}_g, \{7\}_o, \{1, 3, 4, 9\}_1, \{6, 7\}_6, \{4, 8\}_8$
$[\texttt{G}, \texttt{O}, 1..5, 8, 9] \rightarrow \texttt{N}, [6] \rightarrow \texttt{L}$	10	$\{2\}_g, \{7\}_o, \{1, 3, 4, 9\}_1, \{6, 7, 10\}_6, \{4, 8\}_8$
$[\texttt{G}, \texttt{O}, 1..10] \rightarrow \texttt{N}$	11	$\{2\}_g, \{7\}_o, \{1, 3, 4, 9\}_1, \{6, 7, 10\}_6, \{4, 8\}_8$
$[\texttt{G}, \texttt{O}, 1..5, 8, 9, 11] \rightarrow \texttt{N}, [6, 10] \rightarrow \texttt{L}$	12	$\{2\}_g, \{7\}_o, \{1, 3, 4, 9\}_1, \{6, 7, 10, 12\}_6, \{4, 8\}_8$
$[\texttt{G}, \texttt{O}, 1..10, 12] \rightarrow \texttt{N}, [11] \rightarrow \texttt{L}$	13	$\{2\}_g, \{7\}_o, \{1, 3, 4, 9\}_1, \{6, 7, 10, 12\}_6, \{4, 8\}_8, \{11, 13\}_{11}$
Singleton Processing		$\{2\}_2, \{7\}_7, \{1, 3, 4, 9\}_1, \{6, 7, 10, 12\}_6, \{4, 8\}_8, \{11, 13\}_{11}, \{5\}_5$

Table 5: A demonstration of our algorithm using the example in Table 2. The m_j column indicates the index of m_j that the algorithm is currently processing. The first column shows the labels generated for all mention pairs (m_i, m_j), where the indices of m_i are indicated inside the square brackets (e.g, $[\texttt{O}, 1]$ stands for m_o and m_1) and the labels are indicated next to the right arrows (e.g., $\rightarrow$ `L`). The clusters column shows the list of entity sets created by taking the labeling information from the first column.

Table 5 depicts how this algorithm finds the referent relations for all the mentions in Table 2. Note that the special mentions m_g and m_o are considered singular and placed prior to any other mention here. The algorithm labels `L` for (m_g, m_2), which makes *women*$_2 \in C_g$ representing `GENERAL`. For m_4, it labels `L` for m_1 and m_2, which makes *Our*$_4 \in C_1$ representing `JACK`; although *Our*$_4$ is a plural mention, it gets assigned to only one cluster at the moment since the other entity has yet been revealed. For m_7, it labels `L` for both m_o and m_6, which makes *you*$_7 \in C_o$ representing `OTHER` and $\in C_6$ representing `Ross`. For (m_4, m_8), it labels `R` because m_4 is plural and m_8 is singular, which creates a new cluster C_8 and assigns both *Our*$_4$ and *mom*$_8$ to C_8. Once all mention pairs are compared, the algorithm collects mentions that are not assigned to any cluster, and makes them singletons such that *Harmonica*$_5$ becomes the singleton C_5. Furthermore, every mention that belongs to either C_g or C_o gets turned into a singleton such that C_2 and C_7 are created at the end. This is because mentions assigned to those ambiguous types are not referent to one another, if they were, they would have been assigned to `GENERIC` instead.

4.2 Learning Model

Our learning model uses a modified version of the Agglomerative Convolutional Neural Networks (ACNN) introduced by Chen et al. (2017). This architecture incorporates multiple sets of features and learns the most optimized feature combination at each convolution layer. It also allows the model to dynamically accumulate the most salient features for eventual inclusion in the mention and mention pair embeddings. ACNN takes a mention pair (m_i, m_j), performs multiple convolutions to extract features from different groups (CONV$_1$), combines the extracted features among groups using more convolutions (CONV$_2$), and generates mention embeddings $r_s(m_i)$ and $r_s(m_j)$. These mention embeddings are then concatenated with discrete features $\phi_d(m)$ and combined through convolutions to generate a mention-pair embedding $r_p(m_i, m_j)$. The mention-pair embedding together with pairwise features $\phi_p(m_i, m_j)$ are used to make a binary classification for m_i and m_j being referent or not.

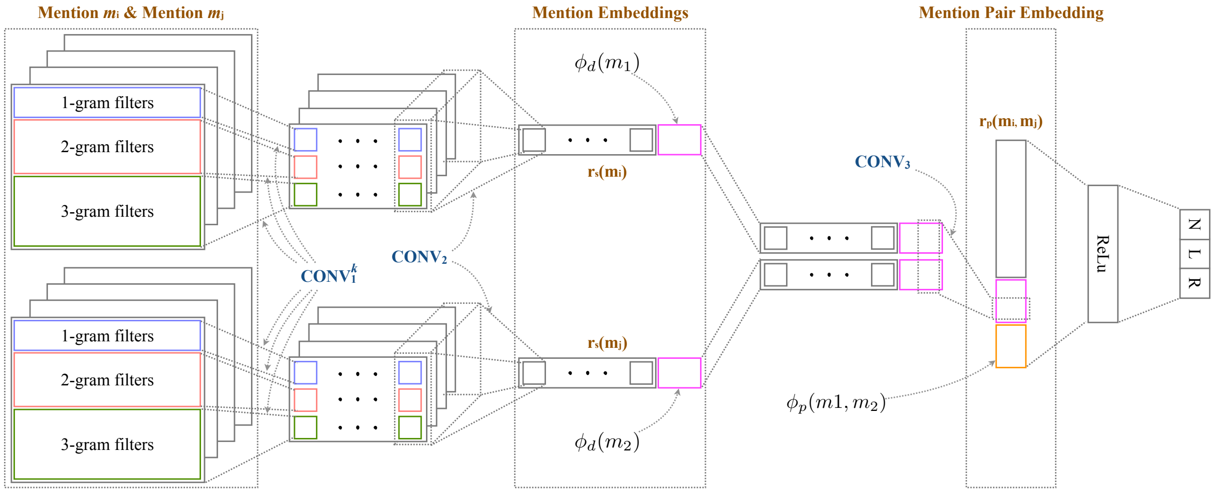

Figure 1: The overview of our coreference resolution model using the multi-class ACNN.

To be adapted to our coreference resolution algorithm in Section 4.1, ACNN is modified at the output layer to include three labels, `N`, `L`, and `R`, such that it is optimized for multi-class instead of binary classification. The modified ACNN, called the multi-class ACNN, generates mention embeddings, $r_s(m_i)$ and $r_s(m_j)$, as well as mention pair embeddings, $r_p(m_i, m_j)$, which are used to create cluster embeddings and fed as input to our entity linking model in Section 5.1.

4.3 Evaluation Metrics

Three metrics proposed by the CoNLL'12 shared task (Pradhan et al., 2012), B^3, CEAF$_{\phi 4}$, and BLANC, are used to evaluate our coreference resolution models. B^3 (Bagga and Baldwin, 1998) is a mention-based metric that measures precision (P) and recall (R) as follows (D: a set of documents, N: the total number of mentions in D, $C_m^{s/o}$: the cluster from the system (s) or the oracle (o) that the mention m belongs to):

$$P = \frac{1}{N} \sum_{d \in D} \sum_{m \in d} \frac{|C_m^s \cap C_m^o|}{|C_m^s|} \qquad R = \frac{1}{N} \sum_{d \in D} \sum_{m \in d} \frac{|C_m^s \cap C_m^o|}{|C_m^o|}$$

In our case, each mention can be assigned to more than one cluster; thus, C_m^* is replaced by the union of all clusters that the mention m belongs to, which enables this metric to evaluate plural mentions.

CEAF$_{\phi_4}$ (Luo, 2005) is an entity-based metric that first creates a similarity matrix $M \in \mathbb{R}^{|S| \times |O|}$ where S and O are the sets of clusters produced by the system and the oracle, respectively. It then measures the similarity between every pair of clusters $(C^s, C^o) \in S \times O$ where $s \in [1, |S|]$ and $o \in [1, |O|]$ such that:

$$M_{s,o} = \frac{2 \times |C^s \cap C^o|}{|C^s| + |C^o|}$$

Given this similarity matrix, the Hungarian algorithm is used to find the list $\mathcal{H}$ that contains similarity scores from the most similar matching pairs of clusters $(C^s, C^o) \in S \times O$ such that $|\mathcal{H}| = \min(|S|, |O|)$. Finally, the overall similarity between S and O is measured as $\Phi = \sum_{\phi \in \mathcal{H}} \phi$, and precision and recall are measured as $P = \Phi/|S|$ and $R = \Phi/|O|$, respectively. Since CEAF$_{\phi_4}$ is entity-based, the metric can be used to evaluate plural mentions without any modification. The potential pitfall is that certain clusters may include a greater number of plural mentions than singular mentions, in which case, distinct clusters with similar sets of plural mentions may yield a high similarity score. However, plural mentions make up less than 10% of the dataset, so we are not concerned about these plural-majority clusters, since most if not all clusters would be dominated by singular mentions.

BLANC (Recasens and Hovy, 2011) is a link-based metric. Let L_s and L_o be the sets of links generated by the system (s) and the oracle (o), respectively. Let G be the set of all possible links between every pair of mentions whether or not they are referent. This metric first creates a confusion matrix $B \in \mathbb{R}^{2 \times 2}$ such that $B_{0,0} = |L_s \cap L_o|$, $B_{0,1} = |L_o - L_s|$, $B_{1,0} = |L_s - L_o|$, and $B_{1,1} = |(G - L_s) \cap (G - L_o)|$. It then measures precision and recall for referent links (P_c and R_c) and also for non-referent links (P_n and R_n):

$$P_c = \frac{B[0,0]}{B[0,0] + B[1,0]} \quad R_c = \frac{B[0,0]}{B[0,0] + B[0,1]} \quad P_n = \frac{B[1,1]}{B[1,1] + B[0,1]} \quad R_n = \frac{B[1,1]}{B[1,1] + B[1,0]}$$

$$F1_c = \frac{2 \times P_c \cdot R_c}{P_c + R_c} \qquad F1_n = \frac{2 \times P_n \cdot R_n}{P_n + R_n}$$

Finally, precision, recall, and F$_1$-score are measured as $P = {P_c + P_n}/2$, $R = {r_c + r_n}/2$, and $F_1 = {F1_c + F1_n}/2$. Note that we decide to replace MUC (Vilain et al., 1995), another popular metric used by the CoNLL'12 shared task, with BLANC because both are link-based and BLANC takes singletons into consideration, which consume a large portion of our dataset (over 20%), whereas MUC does not so that BLANC is more appropriate for our case. It is worth mentioning that a separate confusion matrix B_d is constructed for each document d such that $B = \sum_{d \in D} B_d$ where B_d is based on links only in d. This prevents potential inflation of $B[1,1]$, which could become huge if it were to be measured across the entire dataset. BLANC can also be used to evaluate plural mentions without any modification because each link is treated independently regardless of its mention type in this metric.

5 Entity Linking

5.1 Multi-Task Learning

The task of character identification requires each mention to be identified by the names of actual characters (e.g., `Monica`, `Ross` in Table 2). Figure 2 gives the overview of our entity linking model, which adapts the underlying architecture from the entity linking model proposed by Chen et al. (2017) and generalizes it to jointly handle singular and plural mentions. It assumes the output from ACNN in Section 4.2 such that for each mention m_i, the embedding of that mention and the set of clusters $\{C_1, \ldots, C_k\}$ that m_i belongs to are taken. For each cluster C_a, ACNN gives the list of mention pair embeddings $m_{i,j}^{C_a}$, where $m_i, m_j \in C_a$. Similarly to the previous model, the cluster embedding and the cluster pair embedding are created. Unlike the previous model, our model creates multiple cluster and cluster pair embeddings when m is assigned to more than one cluster during coreference resolution so that the average vectors of those embeddings are generated, which get concatenated with the mention embedding of m_i and passed onto the fully-connected layers for prediction.

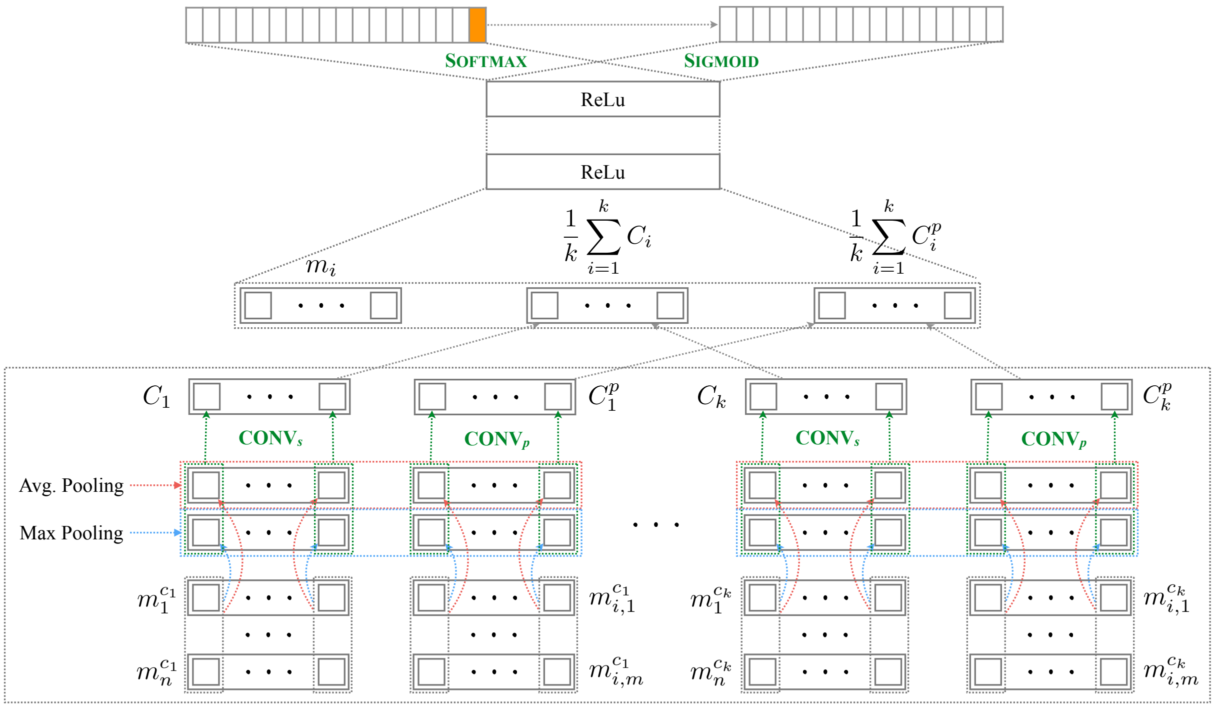

Figure 2: The overview of our entity linking model using multi-task learning.

The final ReLu layer is fed to two output layers optimized by softmax and sigmoid functions, respectively. The dimension of the output layer from softmax is $|E| + 1$ where E is the set of all entities such that each cell represents an entity and the extra cell gives an indication of m being plural. When this extra cell is predicted, the output layer from sigmoid is used, whose dimension is $|E|$, to predict multiple entities for m. Since the sigmoid function optimizes each cell to be between 0 and 1, any entity whose score is greater than 0.5 is taken. These two output layers are optimized jointly, treating the resolution of singular and plural mentions as multi-task learning.

5.2 Evaluation Metrics

Two metrics are used to evaluate the entity linking models. One is the micro-average F1 score whose precision (P) and recall (R) are measured as follows (D: a set of documents, $E_m^{s/o}$: the set of entities found for m by the system (s) or the oracle (o)):

$$P = \frac{\sum_{d \in D} \sum_{m \in d} |E_m^s \cap E_m^o|}{\sum_{d \in D} \sum_{m \in d} |E_m^s|} \qquad R = \frac{\sum_{d \in D} \sum_{m \in d} |E_m^s \cap E_m^o|}{\sum_{d \in D} \sum_{m \in d} |E_m^o|}$$

The micro-average F1 tends to weigh more on frequently occurring entities so it is useful if you need to know the raw prediction power of your model. The other is macro-average F1 score that measures the micro-average F1 for each entity e, say F_1^e, and takes the average, that is $^1/_{|E|} \sum_{e \in E} F_1^e$ where E is the set of all entities. The macro-average F1 treats all entities evenly so it is useful if you need to optimize your model to make correct predictions for as many entities as possible.

6 Experiments

6.1 Configuration

Experiments are conducted on two tasks, coreference resolution and entity linking. For both tasks, models from Chen et al. (2017) are used to establish strong baseline (CZC). Since they take only singular mentions, a pseudo-singular dataset is created where exactly one entity is chosen for each plural mention based on the closest matching previous speaker or if there is none, chosen randomly. Thus, the models trained on this pseudo-singular dataset always predicts one entity per mention. These models are compared to our models described in Sections 4 and 5 (Ours). Additionally, CZC models are evaluated on the singular-only dataset (S-only) where all plural mentions are filtered out, which should give an intuition of how much impact the addition of plural mentions has on the predictions for singular mentions. All results reported from these experiments are averages of three randomly initialized trials. The corpus in Section 3 is split into training, development, and evaluation sets, where all models are tuned on the development set and the best models are tested on the evaluation set. Episodes 1–19, 20–21, and the rest from each season are used to generate the training, development, and evaluation sets, respectively.

6.2 Coreference Resolution

Table 6 shows that our coreference model is capable of learning to handle plural mentions effectively while significantly outperforms the CZC model. The CZC model is trained on the pseudo-singular dataset but evaluated on the full dataset by the metrics adjusted for plurals (Section 4.3) such that it is penalized for not predicting multiple entities for plural mentions. Both the B^3 and BLANC metrics show a similar trend that the CZC and our models achieve higher precision and recall, respectively, whereas our model dominates both precision and recall for the CEAF$_{\phi_4}$ metric. The remarkable gap in performance between these two models signals that our model finds referents for plural mentions well without compromising its ability to find referents for singular mentions. The S-only model gives comparable performance as the one reported by Chen et al. (2017), ensuring that our implementation of the CZC model is robust.

	B^3			**CEAF$_{\phi_4}$**			**BLANC**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
CZC	**84.5±0.6**	60.7±0.2	70.6±0.3	49.0±0.8	63.7±0.3	55.4±0.6	**81.2±1.0**	73.3±0.4	75.9±0.5
Ours	83.8±1.5	**67.0±2.7**	**74.4±1.1**	**52.1±1.2**	**68.0±0.6**	**59.0±0.5**	80.4±0.8	**76.5±1.2**	**78.0±0.6**
S-only	84.3±1.2	71.9±1.4	77.6±1.0	54.5±1.3	71.8±1.0	62.0±0.6	84.3±1.6	80.4±1.1	82.1±1.3

Table 6: Coreference resolution results on the evaluation set (±: standard deviation).

6.3 Entity Linking

Tables 7 and 8 show the micro and macro average scores achieved by all models. For the micro average, the trend is clear across all types of mentions such that the CZC and our models achieve higher precision and recall, respectively. The precision gap for micro average is quite small, signaling that there is no significant loss of ability in entity resolution for singular mentions in our model. For the macro average, our model completely dominates except for the precision of plural mentions, which implies that our model is more generalizable across different entities regardless of their frequency rates in the training set. The recall of micro-average for plural mentions shows relatively high standard deviations for our model. We expect that running more trials of experiments potentially mitigates this variance, which we will explore. It is expected for the micro average scores to be higher than the macro average scores because the micro average favors frequently appearing entities such that it is possible to achieve high micro average scores without handling infrequent entities well, whereas that is not the case for the macro average.

	Singular			**Plural**			**All**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
CZC	**72.8±0.5**	72.8±0.5	72.8±0.5	**60.8±2.4**	19.7±0.8	29.8±1.2	**71.8±0.4**	61.4±0.4	66.2±0.4
Our	72.7±0.3	**72.9±0.4**	**72.8±0.4**	59.9±1.7	**32.2±4.8**	**41.7±4.1**	71.1±0.4	**64.2±1.3**	**67.4±0.8**
S-only	73.7±0.6	73.7±0.6	73.7±0.6						

Table 7: Micro-average scores for entity linking on the evaluation set (±: standard deviation).

Table 9 shows the micro average F1 score for each entity. The top-15 frequently appearing characters are considered to be known entities, whereas all the other secondary characters are considered `OTHER`, which composes about 26.8%. Our model dominates all the main characters (the first six entities) and `OTHER`, together of which gives about 90% of the entire annotation. Given that these results are achieved by using automatically generated clusters from our coreference resolution models, they are encouraging.

	Singular			Plural			All		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
CZC	72.9±5.0	55.5±1.0	59.4±2.3	**37.9±1.0**	10.5±0.3	14.0±0.3	71.1±4.6	46.2±1.1	53.2±1.9
Our	**75.8±1.4**	**56.9±1.1**	**61.8±1.1**	34.8±5.0	**15.8±1.7**	**20.5±1.6**	**74.2±1.4**	**48.8±1.5**	**55.5±0.8**
S-only	73.3±2.5	55.4±1.6	59.6±2.3						

Table 8: Macro-average scores for entity linking on the evaluation set (±: standard deviation).

	Ro	**Ra**	**Ch**	**Mo**	**Jo**	**Ph**	**Em**	**Ri**	**Ca**	**Be**	**Pe**	**Ju**	**Ba**	**Ja**	**Ka**	**OT**	**GN**
CZC	69.2	77.5	69.0	71.3	71.5	79.0	**63.4**	76.4	**31.3**	**41.8**	**56.4**	09.3	**49.2**	11.8	24.7	58.2	**45.1**
Our	**71.9**	**78.4**	**71.5**	**72.2**	**72.3**	**79.7**	61.5	**82.0**	29.6	**41.8**	54.8	**12.8**	45.0	**18.2**	**47.3**	**59.2**	**45.1**
S-only	78.3	86.5	78.8	81.7	78.3	88.8	69.2	83.9	40.3	39.3	59.2	16.1	39.8	24.8	35.2	64.0	49.7
%	12.65	11.58	11.16	9.71	9.33	8.61	0.98	0.96	0.71	0.64	0.57	0.44	0.34	0.28	0.26	26.79	5.01

Table 9: Entity linking results on evaluation set per character.Ro: Ross, Ra: Rachel, Ch: Chandler, Mo: Monica, Jo: Joey, Ph: Phoebe, Em: Emily, Ri: Richard, Ca: Carol, Be: Ben, Pe: Peter, Ju: Judy, Ba: Barry, Ja: Jack, Ka: Kate, OT: `OTHER`; GN: `GENERAL`.

7 Conclusion

In this paper, we explore a new paradigm for handling plural mentions in two resolution tasks, coreference resolution and entity linking, on multiparty dialogue. We address this challenge by showing the inadequacy of traditional approaches in handling plural mentions, and present an innovative approach to overcome the shortcomings of existing methods for these tasks at hand. For resource creation, we expand upon the Character Identification corpus and augment it with the manual annotation of plural mentions (Section 3). For linguistic analysis, we propose a novel transition-based algorithm and evaluation metrics to process different types of mentions for coreference resolution (Section 4). For NLP engineering, we introduce a neural-based entity linking model using multi-task learning that comprehensively handles plural mentions (Section 5). The results of our models demonstrate significant improvements on these tasks, implying the feasibility of our approach to handle plural mentions (Section 6).

To the best of our knowledge, this paper provides the first extensive framework for resolving referents for plural mentions, which is a critical problem in any resolution task. Further work includes improving the quality of the dataset as well as expansion of its size, and addressing the issue of extracting global and external features for complete coreference and entity resolution for both singular and plural mentions. All resources including the annotated corpus and source codes are publicly available through the Character Identification project: `https://github.com/emorynlp/character-identification`.[5]

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*, pages 563–566.

Henry 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*, SIGDIAL'16, pages 90–100.

Henry Yu-Hsin Chen, Ethan Zhou, and Jinho D. Choi. 2017. Robust Coreference Resolution and Entity Linking on Dialogues: Character Identification on TV Show Transcripts. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, CoNLL'17, pages 216–225, Vancouver, Canada.

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*, EMNLP'16, pages 2256–2262.

[5] The Character Mining project provides a superset of the corpus presented in this paper for several other tasks: `https://github.com/emorynlp/character-mining`.

Greg Durrett, David Hall, and Dan Klein. 2013. Decentralized entity-level modeling for coreference resolution. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 114–124, Sofia, Bulgaria, August. Association for Computational Linguistics.

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*, pages 1256–1261, San Diego, California, June. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark, September. Association for Computational Linguistics.

Prateek Jain, Manav Ratan Mital, Sumit Kumar, Amitabha Mukerjee, and Achla M. Raina. 2004. Anaphora resolution in multi-person dialogues. In Michael Strube and Candy Sidner, editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 47–50, Cambridge, Massachusetts, USA, April 30 - May 1. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September. Association for Computational Linguistics.

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

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.

Marta Recasens and Eduard H. Hovy. 2011. BLANC: Implementing the Rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

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*, MUC6 '95, pages 45–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, pages 994–1004, San Diego, California, June. Association for Computational Linguistics.