Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Ethan Zhou March 31, 2018

A Thesis on Character Identification

by

Ethan Zhou

Jinho D. Choi

Adviser

Math and Computer Science Department

Jinho D. Choi

Adviser

Susan Tamasi

Committee Member

Lee Cooper

Committee Member

2018

A Thesis on Character Identification

By

Ethan Zhou

Jinho D. Choi

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Math and Computer Science Department

2018

Abstract

A Thesis on Character Identification
By Ethan Zhou

Traditional coreference resolution systems use methods insufficient for completely resolving plural mentions, especially when applying conventional coreference concepts to different tasks such as character identification. This paper gives a comprehensive view of one of the least examined yet most difficult parts of entity resolution–particularly coreference resolution and entity linking. Since our approach to entity resolution focuses on its applicability to character identification, we use the character identification corpus from SemEval 2018 and expand the dataset in scope to include plural mention annotations. We then show the inadequacy of these concepts and show an innovative design to overcome the shortcomings of traditional coreference ideas for the character identification task in this paper. Our innovative design includes an all-new algorithm for coreference resolution that selectively creates clusters to handle all types of mentions, singular and plural, as well as a new joint deep learning approach to entity linking determine the entities for both singular and plural mentions as well. Using our novel design, we demonstrate that our coreference and entity linking models surpass more traditional models. To the extent of what we know, we are the first to extensively investigate plural mentions in the context of entity resolution.

A Thesis on Character Identification

By

Ethan Zhou

Jinho D. Choi

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Math and Computer Science Department

2018

Table of Contents

A Thesis on Character Identification

Ethan Zhou
Emory University
ethan.zhou@emory.edu

1 Introduction

Character identification is a relatively new task proposed by Chen and Choi (2016) and while it can be considered a novel task, it is in fact an application of old concepts to a new field: coreference resolution and entity linking of a multi-party conversation. These entity resolution tasks are highly complex since their scopes typically encompass entire documents or even multiple documents. They are already challenging considering that many of the state-of-the-art systems (Clark and Manning, 2016; Francis-Landau et al., 2016; Wiseman et al., 2016; Gupta et al., 2017; Lee et al., 2017) still show only moderate improvement even though these systems focus more on singular mentions than on plural mentions and plural mentions make up a significant portion of all the mentions in a document(s).[1] In fact, coreference resolution remains as one of the last unresolved fundamental tasks of natural language processing, and few have even considered dealing with plural mentions in a comprehensive manner. Table 1 illustrates the differences in annotation for coreference resolution between the CoNLL'12 shared task (Pradhan et al., 2012) and our proposed solution. In the CoNLL'12 annotation, the plural mention $\textit{They}_8$ would be in the same cluster as [$\textit{Mary}_1$ and $\textit{John}_2$]$_3$, but the other plural mention $\textit{We}_7$ forms a singleton cluster because there is no other noun phrase which refer to the exact same entities. Although it is quite obvious that while $\textit{They}_8$ and $\textit{We}_7$ do not refer to the exact same set of entities and are most definitely not directly coreferent, they are indirectly coreferent since these two plural mentions have a common subset of referent entities. This limitation in the CoNLL'12 annotation methods forces the gold data to drop

[1] We define a singular mention to be a noun phrase which refers to one entity and a plural mention to be a noun phrase which refers to more than one entity.

some important coreference links to individual entities that need not necessarily be thrown away.

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, we maintain these links to individual entities by linking the plural mentions *We*$_7$ and *They*$_8$ to each entity that those mentions refer to. Because coreference resolution is such a fundamental task in natural language processing, enabling models to learn these links to individual entities may positively affect higher-level tasks such as question answering or machine translation. However, there is a trade-off: entity resolution tasks–coreference resolution and entity linking–now have added complexity, which has not been directly addressed yet. In this paper, we address the lack of a solid foundation for coreference of plural mentions by resolving important issues that arise when dealing with them. First, we propose an annotation framework for plural entity resolution, which we use to expand the character identification corpus (Section 4). We completely redo the algorithm by Chen and Choi (2017) to enable the system to link to both antecedents and postcedents as well as allowing each mention to link to more than one mention when necessary, and we adapt some evaluation metrics to handle plural mentions as well (Section 6). We also provide a novel joint deep learning entity linker which identifies both singular and plural mentions (Section 7). For evaluation, we run our models on our dataset generated by the character identification corpus (Section 8), and our experiments reveal improvement from our new models compared to a previous state-of-the-art model dedicated to singular mentions. As far as we can tell, we are the first to deploy extensive annotations for plural mentions on a large scale; in fact, we are able to develop and use an annotation framework and deep learning models for this task to achieve promising results for plural mention resolution.

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. Character identification is a task that requires systems to identify the entities, who may not necessarily be an active member in the conversation, for each personal mention in a conversation or dialogue. Our task is different from traditional entity linking tasks like Wikification because character identification focuses specifically on dialogues where the entities are present in either the main cast or in the supporting cast. Chen et al. (2017) later enhanced the annotation framework by introducing ambiguous entity types as well as making qualitative improvements to the *Friends*-based corpus. We continue to use the annotation framework and the corpus for our work, and we expand upon the corpus by annotating two additional seasons of Friends, with all annotated seasons replete with singular and plural mention annotations.

Since our annotations are designed to support both coreference resolution and entity linking, we use our character identification corpus for both tasks. We were partially motivated to confront plural resolution because previous works that dealt with general cases of coreference resolution–such as Clark and Manning (2016) and Durrett et al. (2013)–did not handle plurals to our satisfaction. Their approaches used the CoNLL'12 corpus, which as we have explained before in Table 1, do not provide a meticulous study of plural mentions. There has been one work by Jain et al. (2004) which provided a rule-based system for resolving plural mentions, but their system was limited to plural types with a known number of entities. We distinguish our approach by handling both plural types with a known number of entities and with an unknown number of entities, making it more challenging. We also control the full stack, from annotation to coreference to entity linking. We are also inspired in design and approach by Chen et al. (2017); they presented a character identification approach for singular mentions from a coreference system to an entity linking model that identified the character referents singular mentions. We adapt this approach to develop a new multi-task learning model that jointly handles singulars and plurals in this paper.

3 Background

Our task falls into the field of natural language processing, in which we use computational methods to enable computers to learn language patterns. In particular, we use deep learning methods because neural networks offer superior performance in almost every aspect compared to other machine learning techniques. Moreover, before the popularization of neural networks, gaining insight into spoken and written language proved to be a challenge since words could not be easily translated into numerical representations. Now generating numerical representations of words can be easily and quickly done with a neural network (Mikolov et al. (2013)), which in turn also makes deep learning in higher-level tasks not only a viable choice but the preferred one.

For our computational methods, we use custom built neural networks to tailor our approaches specifically to our task of character identification. The backbone of our deep learning approaches uses two libraries specifically designed to build deep learning models: TensorFlow (Abadi et al. (2016)) and Keras (Chollet and others (2015)). The library incorporates a variety of neural techniques, and for the purposes of this paper we give a basic explanation of the two major ones we use in our research:

1. Feedforward layer - the simplest neural layer. It is an array of perceptrons (think neurons) which feeds all output forward to the next layer. Also called a dense layer, since every perceptron from a dense layer connects to every perceptron in the second layer.

2. Convolutional layer - the ”visual” neural layer. It was originally used for visual image processing, but has since been adapted to natural language processing. Convolutional layers pulls out the most salient features from the input in a condensed form. They are mainly useful in natural language processing when extracting salient features from groups of words, phrases, sentences, etc. since a group of linguistic elements can be restructured into a matrix, where each row represents one linguistic element, and can be viewed as an ”image”.

4 Corpus

We curate our data from a special source, the Friends TV show transcript, because it offers a glimpse into realistic English conversations and has been made publicly available to anyone through fans of Friends who meticulously copied down the transcript in as detailed a manner as possible. [2] In fact, many English-as-second-language speakers often use Friends as a way to learn natural English instead of the stiff formalities that are drilled into one's head through rote memorization in class. This verifies our belief that Friends transcripts offer a special insight into multi-party conversational dialogue, which few other datasets offer despite that conversational dialogue makes up the vast majority of a language's usage.

4.1 Definitions

Before diving into the details, we would like to provide a few definitions for clarity. We define a mention to be a noun phrase which refers to a specific entity or object. For our purposes, we care only about the mentions which refer to a specific living entity; all other mentions are ignored. We define a referent to be a character/entity in the Friends TV show, named or unnamed, to which each mention is linked. We also define a singular mention to be one that refers to only one entity while a plural mention is a mention which refers to multiple entities.

4.2 Data

For our data, we use the character identification corpus by Chen and Choi (2016), which specializes in referent annotations of mentions referring to the characters in the show. Although the corpus has been vastly improved since its inception, all annotations were primarily geared towards resolving singular mentions. While perfectly acceptable for previous character identification models, the lack of annotations for plural mentions hindered our current task of plural mention character identification. As such, we expanded the dataset using crowdsourcing techniques with two additional seasons, bringing the total number of seasons in the dataset to four, replete with annotations for both singular and plural mentions.

[2]Friends transcripts: `http://www.livesinabox.com/friends/scripts.shtml`

4.3 Schema

Before any real data annotation can occur, we must provide a set of annotation labels with which each mention can be classified. Chen et al. (2017) already provide extensive groundwork for annotating singular mentions, but since we are handling plural mentions, the labeling scheme must be slightly altered. Previously, Chen et al. (2017) use three sets of labels: primary, secondary and ambiguous labels. Primary labels are simply the names of the six primary characters of Friends: Monica Geller, Ross Geller, Rachel Green, Chandler Bing, Phoebe Buffay and Joey Tribbiani. Secondary labels are simply the names of secondary characters (i.e. any characters with some sort of unique identifier). Examples include Rachel's parents, Sandra and Leonard Green, as well as Monica and Ross' parents, Judy and Jack Geller. Ambiguous labels are simply labels that are used to signify that the referent of the mention either cannot be specifically identified or does not exist. There are three main subcategories for ambiguous labels: Other, General, Generic. These will be explained in further detail in Section 4.4. We also introduce a new label, Non-Entity, to differentiate between mentions which do not reference characters, but rather objects. Such a label helps filter out mentions which are unrelated to our task.

4.4 Annotation

The character identification corpus has been collected into one large repository called the Character Mining project, which includes all ten seasons of the *Friends* TV show transcripts.[3] It includes the annotations of the first two seasons by Chen et al. (2017), which is publicly available through the International Workshop on Semantic Evaluation (SemEval 2018)[4], as well as our most recent additions of seasons three and four. In our recent additions, we made the following changes to the corpus:

1. Annotations from Chen et al. (2017) were incomplete for the first and second seasons. We completed the annotation for the first two seasons and annotated two more seasons using guidelines similar to ones used previously, bringing the total number of annotated seasons to

[3]Character Mining: `https://github.com/emorynlp/character-mining`
[4]SemEval 2018 Task 4: `https://competitions.codalab.org/competitions/17310`

four.

2. Speaker and the entity labels were mismatched in the previous annotation, so we standardized all names across the transcripts to be full names. 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`. While a seemingly benign error, computers do indeed still need standardization to understand that `Monica` is equivalent to `Monica_Geller`.

3. We used two rounds of annotations to annotate plural mentions. The first round identified the plural mentions, and the second round annotated the plural mentions. We used the `COLLECTIVE` annotation type to distinguish plural mentions in the previous annotation, but we discarded it in favor of deterministically distinguishing between singular and plural mentions by the size of its entity set.

We use the same annotation guidelines for both singular and plural mentions since the annotations can be easily adapted for plural mentions without affecting the singular mentions by simply annotating the plural mentions with the number of entities to which each plural mention refers. Thus, we can say that entity e falls into one of the four groups below, for each entity e in the set of entities E which annotate each mention m:

1. Known entities: main cast and recurring support cast in the show.

2. `GENERIC`: characters in the support cast of the show whose identities are never revealed:
   e.g., That *waitress* is really cute, I am going to ask *her* out.

3. `GENERAL`: any mention that uses said mention to represent a class of people rather than a specific character:
   e.g., The ideal *guy* you look for doesn't exist.

4. `OTHER`: characters in the show whose identities are currently unknown from local context but can be inferred from the entire show.

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 Geller` $\{Our_4, mom_8\} \rightarrow$ `Judy Geller`,
$\{Harmonica_5\} \rightarrow$ `Monica Geller`, $\{Ross_6, you_7, I_{10}, me_{12}\} \rightarrow$ `Ross Geller`,
$\{women_2\} \rightarrow$ `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 gives examples of the annotation labels we use to label singular and plural mentions. The mention *women*$_2$ does not refer to any specific character so it is identified as `GENERAL`. We can see that two mentions, *guy*$_{11}$ and *He*$_{13}$, refer to the same character, but since he is a minor support character and makes a fleeting appearance, we do not find out his identity, so we label these mentions with the generic type, `MAN_1`. For the plural mention *Our*$_4$, it is quite obvious that the referents of this mention are `Jack` and `Judy`, so we annotate this mention with labels `Jack Geller` and `Judy Geller`. Unfortunately, the same could not be said for the plural mention *you*$_7$. We can only identify one of the referents for *you*$_7$: `Ross Geller`. However, it is quite obvious that this mention is referring to more than one person, so we use `OTHER` as a way to fill the gap, so that we know it refers to someone else but not necessarily whom or how many. We use this method is used to prevent confusion of the non-immediately identifiable entities with generics like `MAN_1`.

4.5 Crowdsourcing

Annotating the Friends transcripts proved to be more than one person can handle, so we opted for a less time-consuming approach: crowdsourcing. Because training deep learning neural networks requires large quantities of data, crowdsourcing has become the norm for creating large datasets owing to its good balance of time and quality: it saves time through parallelization and returns decent quality when properly controlling for bad elements. Moreover, because crowdsourcing has become such an indispensable tool, Amazon has created a crowdsourcing marketplace–called Amazon Mechanical Turk or MTurk–where simplicity and speed are paramount, allowing us to post annotation tasks and review results in real time.

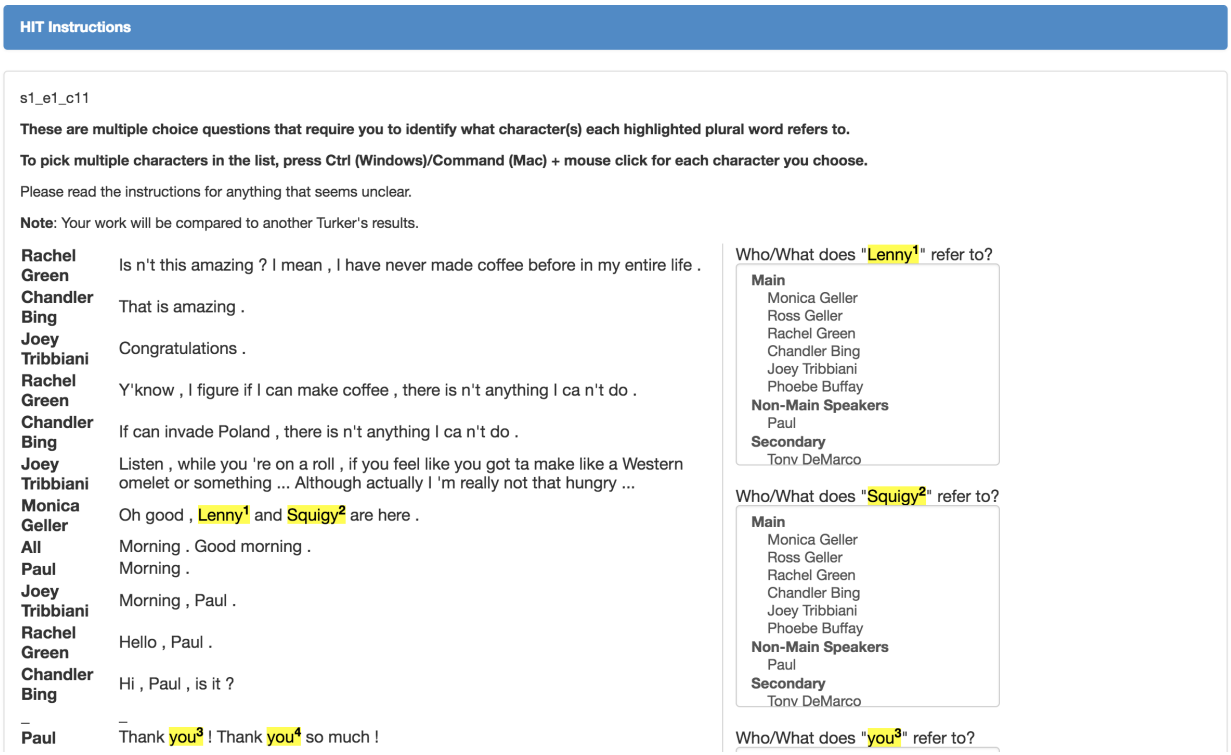

Figure 1: Example of MTurk scene assignment. Also called HIT.

MTurk is organized to allow two parties: requesters and workers. The parties behaviors are quite obvious: requesters request work to be done and the workers choose whether or not to do the work. As requesters, we must design an intuitive interface with clear instructions to allow workers to maximize their time completing the work rather than decoding what the task is.

To meet these requirements, we divide the Friends transcripts into tiers: seasons are composed of episodes, episodes are composed of scenes, scenes are composed of utterances. Seasons and episodes are quite obvious since the TV show is naturally organized in such a manner, but we divide episodes into scenes because it is easier for the workers to read. Every scene counts as one assignment and is paired with a set of questions which ask for the referent of every mention detected within the scene, which requires the worker to read each scene to be able to answer the questions. We then upload multiple batches of these scene assignments to MTurk to take advantage of multiple workers simultaneously completing our assignments. An example of one of these assignments is shown in Figure 1.

4.6 Quality Control

While MTurk overall provides benefits of good quality for most users, we happen to run across some trouble with our task. Because of limitations of heuristic filters to remove poorly annotated data, we manually curated the data through identification of the good workers and subsequent rejection of bad workers. It obviously follows that while this remains a viable technique, our approach to reviewing data annotation is not exhaustive, which results in potentially erroneous data in some sections.

We also note that our corpus is not as well-formed as we expected it to be. A plural mention should refer to more than one entity while a singular mention should refer to only one entity, which means the only supposed difference between them are the number of entities to which the mentions refer. In reality, the difference is more muddied because some plural mentions refer to only a singular entity and other plural mentions only have one referent, which can happen because workers do not correctly identify all the referents of the plural mention.

It is also curious to note that we have observed through manual evaluation of worker annotations that they are far from perfect in annotating the mentions correctly, singular or plural. And although we do not have hard evidence for this observation, it seems that workers spend more time on identifying plural mentions than on identifying singular mentions and are producing less accurate results. Whether from lack of motivation or true inability to infer the real entity referents, we cannot say for sure, but we believe that workers are simply having a harder time with plural mentions given the scene context, which is something we did not expect. If this belief holds true, then it must mean that character identification requires global and external knowledge about the characters, the conversational context and the show itself to resolve this task.

4.7 Analytics

Table 3 breaks down our corpus transcript-wise, mention-wise and label-wise. We more than double the size of the annotation; Chen et al. (2017) measured 18,608 mentions in the previous iteration of this dataset, while the expanded corpus is comprised of 47,367 annotated mentions, approximately

increasing the size by a factor of 2.5. Plural mentions make up about 9% of the entire current dataset, a significant enough portion to influence the entity resolution tasks. We treat each scene as an independent dialogue for technical purposes.

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.

We require that all mentions be annotated by two workers. The corpus has a Cohen's kappa score of 56.88% for plural mentions, making it approximately 20% lower than the score for singular mentions (Chen and Choi, 2016). Initially, we held this requirement as a rollover from the guidelines for singular mention annotations to control for poor annotations, but we eventually realized that this requirement served another task: filling in incomplete annotations. Because of the complexity of this task, we took the union of the annotations by both workers for each question as the answer because sometimes both workers would give complementary answers, effectively giving a complete or almost complete set of entities for many plural mentions, to enhance the annotations.

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.

Table 4 shows the distributions of entity types. Our dataset consists of 67% mentions with primary characters as referents with only 8.6% ambiguous mentions, suggesting that the vast majority of

mentions have known entity referents. It is also curious 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. It would seem that these two ambiguous entity types are easily confused by the MTurk workers since these are ambiguous labels and possibly because our instructions were not absolutely clear on what each ambiguous label refers to. We expect to investigate this phenomenon further, but we suspect that the natures of ambiguous labels are not easily separated; for sometimes even we have some trouble making a clear cut decision for certain mentions.

5 Approach

Our approach is twofold: coreference resolution and entity linking. We take inspiration from the approach devised by Chen et al. (2017) and make improvements upon it. The improvements use two different methods for testing: singular training and plural training with plural evaluation on both methods. These two categories shall be further explained in the following paragraphs.

However, we cannot jump the gun just yet, for we must address a few issues for a true understanding of our systems. Our data is far different from any other datasets dealing with coreference resolution or entity linking primarily because the data focuses on a cast of identifiable characters. To be sure, there are certainly unidentifiable ones, and we deal with those in a separate way, which has been talked about in Section 4. As such, plural mentions are identifiable in our dataset. Since plurals can be identified, we come upon a difficulty: traditional coreference systems tend to put plural mentions into clusters separate from the singular mentions even though plural mentions can be referent to the singular mentions. While passable for coreference systems previously, our unique dataset requires a different but intuitive perspective: singular and plural mentions are no longer separated.

But mixing singulars and plurals together significantly increases the complexity of coreference resolution from training to decoding to evaluation. To the human mind, using this method is very intuitive; however, coreference clusters are no longer easily separable since multiple difference coreference clusters can now be linked together with plural mention(s). By enabling this type of

linking, any model that does character identification coreference resolution must implicitly learn to differentiate between singular and plural mentions while also connecting each mention to the correct cluster(s). Already we can see that this idea already starts encroaching upon the topic of mention detection, which in itself is already a challenging task.

Moreover, just simply linking plural and singular mentions to each other is not a good idea because having either singular or plural mentions link to another plural mention creates chaos. Enabling this mechanism begs the question: how will the model know which cluster(s) are being referred? Using this mechanism requires enabling entity linking, which suggests that future coreference resolution systems for character identification require a tight coupling with an entity linking system that can dynamically provide features for coreference during training. And indeed, resolving this problem fully may simply require a system that does not differentiate coreference from entity linking and instead accomplishes both in one shot. For now we just maintain the separation of coreference resolution and entity linking.

6 Coreference Resolution

Introducing plural mentions broadens the scope of coreference resolution; rather than just searching for one coreferent mention, the task requires looking for multiple coreferent mentions rather than just one. This requirement forces us to think differently than traditional coreference methods. Usually the strategy for traditional coreference works in an intuitive manner that creates clusters of mentions in which all mentions within said cluster refer to the same entity and works as such: given a mention m_j, a conventional coreference system search all antecedent mentions to find a mention m_i which is coreferent to m_j and then adds m_j to the mention cluster C_i of m_i; if there is no such coreferent antecedent, the system creates a new cluster which contains only mention m_j.[5] When the system completes this process, it stops searching for additional coreferent mentions for m_j and moves onto the next mention m_{j+1}. While adequate for singular mention resolution, plural mention resolution requires a more refined approach since our approach allows plural mentions to be exist in

[5]We define a cluster to be a group of mentions which all refer to the same entity or entities in a document. Clusters across documents may refer to the same entities.

multiple clusters. Moreover, when using the conventional, referents have transitive relationships, where given three mentions, m_1, m_2 and m_3, if m_1 is coreferent to m_2 and m_2 is coreferent to m_3, then by transitivity, m_1 is coreferent to m_3. This transitive property is lost with the presence of plural mentions in multiple clusters, and we give a formal comparison between the previous approaches and our novel approach: We let $m_i \leftrightarrows m_j$ be a coreferent relation between m_i and m_j, that is m_i and m_j have a direct coreferent link between them. Viewed from the perspective of traditional coreference, $m_i \leftrightarrows m_j$ and $m_j \leftrightarrows m_k$ implies $m_i \leftrightarrows m_k$, as previously stated. However, when applying our approach to plural coreference, $m_i \leftrightarrows m_j$ and $m_j \leftrightarrows m_k$ no longer necessarily mean $m_i \leftrightarrows m_k$ if m_j is a plural mention (though this transitivity property still holds if all three mentions are singular). Then, m_j belongs to two different clusters $C_i = \{m_i, m_j\}$ and $C_k = \{m_j, m_k\}$, and it becomes quite obvious that m_i and m_k are definitely not coreferent. The expansion of coreference resolution to comprehensively handle plural mentions also requires us to tweak the evaluation of our system slightly since some of the popular evaluation metrics for coreference resolution such as B^3 (Bagga and Baldwin, 1998) are not necessarily designed with plural mentions in mind.

Figure 2 is a visual representation of one of the traditional systems that embodies the original strategy for handling coreference resolution–the ACNN by Chen et al. (2017).[6] Because this coreference system was built to specifically target singular mentions, we adapt the architecture slightly to integrate the neural model with our new algorithm, which will be explained in Section 6.1. Both the ACNN and the current coreference model use a mention-pair approach to coreference, where every mention is compared to every previous mention to find the best antecedent(s). We still believe an expansive search strategy approach is a viable one for plural resolution because plurals can be coreferent to multiple mentions, making it all the more crucial to visit as many mentions as possible to make up for any possible weaknesses within the neural model's learning.

We discuss our novel approach to plural mention coreference resolution in Section 6.1. We introduce an algorithm that learns to create clusters depending on the type of coreferent relationship between two mentions. This algorithm still maintains the traditional approach of differentiating

[6]See Chen et al. (2017) for more detail on the ACNN approach to coreference resolution.

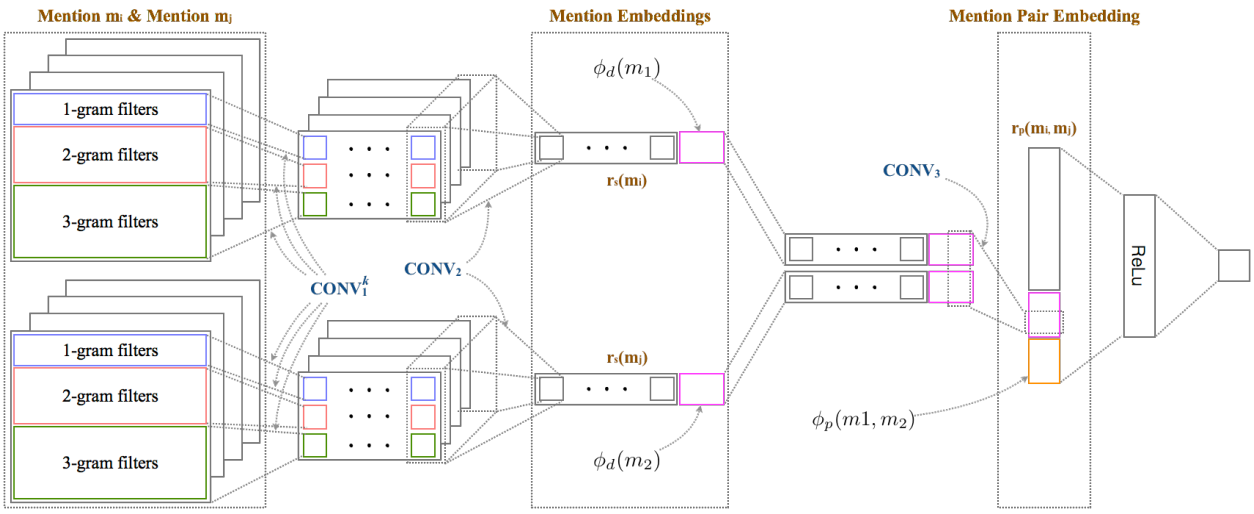

Figure 2: Original coref model by Chen et al. (2017).

between entities by designating non-coreferent singular mentions to different clusters, but is allowed to choose as many clusters as it deems correct for each plural mention. For example, given a plural mention m_p and a singular mention m_i, let $m_p \leftrightarrows m_i$. Our algorithm creates a cluster C_i such that C_i contains both m_p and m_i. Now, given a singular mention m_j which is different from m_i, let $m_p \leftrightarrows m_j$. Our algorithm has two choices: assign m_j to C_i if $m_i \leftrightarrows m_j$ or create a new cluster C_j which contains m_j and m_p if not. This choice cannot be made through rule-based decision-making, so our algorithm teaches this decision to appropriately create clusters to our coreference neural model in training. Though we are able to apply a new algorithm to our new approach towards resolving plural mentions, the existing evaluation metrics must be readjusted to properly evaluate our model on both singular and plural mention coreference resolution. We discuss our alterations to the evaluation metrics in Section 6.2; based on what we know, this is the first attempt to modify coreference metrics to account for plural mentions linked to multiple entities.

6.1 Algorithm

For each mention m_j, our algorithm compares it against every antecedent m_i where $0 < i < j$ to check whether the two mentions are coreferent. For every mention, we add two more pseudo-mentions, m_g and m_o, to compare against m_j. They represent the `GENERAL` and the `OTHER` types, respectively (Section 4.4). If m_j should be considered coreferent to either m_g and m_o, the mention

m_j is automatically put into a singleton cluster by our algorithm. Then, for each mention pair (m_i, m_j), the algorithm assigns one of the following three labels for multi-classification:

1. N: there is no coreferent relation between m_i and m_j.

2. L: assign mention m_j to cluster C_i which contains mention m_i. If C_i does not exist, then create a new cluster for C_i and assign both m_i and m_j to C_i.

3. R: assign m_i to cluster C_j which contains mention m_j. If C_j does not exist, then create a new cluster for C_j and assign both m_i and m_j to C_j.

The algorithm trains the model to learn these labels by generating gold training data. L is labeled if m_i is a singular mention or if m_i is either m_g or m_o. R is labeled if m_i is plural and m_j is singular. N is labeled for all the other cases. This algorithm disallows the plural-plural links although they may be indirectly coreferent by linking to coreferent singular mentions. This precaution prevents clusters comprised of only plural mentions since plural-only clusters do not contain useful information for identifying the correct characters to which the mentions refer. However, it is possible for plural mention singleton clusters to exist since plural mentions can link directly to OTHER and GENERAL to allow for the possibility that the plural mention refers to entities who are unidentifiable at the moment. It is also possible to link plural mentions directly by using the GENERIC type (Section 4.4), which is not adapted to annotate entities for plural mentions in the current annotation scheme.

Table 5 illustrates an example of the algorithm's internal mechanisms and the resulting output when searching for coreferent relations for each mention in Table 2. To allow for comparison to every mention, the algorithm places the special mentions m_g and m_o as antecedents to every mention. Since *women*$_2$ has been annotated as GENERAL, the algorithm assigns the label L to (m_g, m_2) to make *women*$_2$ a singleton cluster. For m_4, it assigns the L coreferent link for mention pair (m_1, m_4) and (m_2, m_4), putting ***Our***$_4$ into C_1, where C_1 represents JACK. Although ***Our***$_4$ is a plural mention, we dynamically designate the plural mention to different clusters since other entity or entities may not yet have been revealed, as it is in this case. For m_7, the L type coreferent link is assigned to

$[m_i] \rightarrow \{$**N, L, R**$\}$	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., $\rightarrow \texttt{L}$). The clusters column shows the list of entity sets created by taking the labeling information from the first column.

both mention pairs (m_o, m_7) and (m_6, m_7), which causes *you*$_7$ to be made into a singleton cluster and puts m_7 into cluster C_6 which represents Ross. For (m_4, m_8), the R type coreferent link is assigned because m_4 is plural and m_8 is singular, which means that a new cluster C_8 based on mention m_8 is created and C_8 now contains both *Our*$_4$ and *mom*$_8$. Any leftover mentions which have not been assigned to a cluster are picked up and made into singleton clusters, so *Harmonica*$_5$ forms a singleton C_5.

We use the same model as Chen et al. (2017), the ACNN, to learn and predict these three labels: N, L, and R by using the same set of features as the previous work. Of course, our model is optimized for 3-labels rather than binary classification. We then use the ACNN to produce mention and mention pair embeddings as features for the entity linking model, which will be described in Section 7.1.

6.2 Evaluation Metrics

We modify our evaluation metrics accordingly to accommodate plural mentions. Some typical metrics used for coreference resolution are three metrics proposed by the CoNLL'12 shared task

(Pradhan et al., 2012): B^3, $CEAF_{\phi_4}$, and BLANC.

B^3 (Bagga and Baldwin, 1998) is a mention-based metric which measures F-measure according to which clusters each mention is assigned to. We provide the B^3 formula below. Given a set of documents D, the total number of mentions N in D, the cluster $C_m^{s/o}$ from the system (s) or the oracle (o) that mention m belongs to:

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

We adapt this formula in the event that some mentions are assigned to multiple clusters; then, C_m^* simply refers to the union of all clusters that contain mention m, which now allows this metric to calculate scores for plural mentions as well.

$CEAF_{\phi_4}$ (Luo, 2005) is an entity-based metric that measures the similarities between system-constructed clusters and oracle-constructed clusters. The metric generates 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, and measures the similarity between every cluster pair $(C^s, C^o) \in S \times O$ where $s \in [1, |S|]$ and $o \in [1, |O|]$. We end up with a formula:

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

The metric then uses the similarity matrix to find the best matching pairs of clusters using the Hungarian algorithm by generating a list $\mathcal{H}$ that contains the highest similarity scores from the best matching pairs of clusters $(C^s, C^o) \in S \times O$, where $|\mathcal{H}| = \min(|S|, |O|)$. The overall similarity score between S and O is measured as $\Phi = \sum_{\phi \in \mathcal{H}} \phi$, and is used to calculate precision and recall: $P = \Phi/|S|$ and $R = \Phi/|O|$. This metric does not need to be modified to be able to evaluate plural mentions. The only potential pitfall is that clusters which include plural mentions may have a greater number of plural mentions than singular mentions, which means that distinct clusters with similar plural mentions may be confused together. However, plurals consist of only 9% the dataset, so we do not have strong concerns about confusion between clusters dominated by plural mentions,

since most if not all clusters will be have a majority of singular mentions.

BLANC (Recasens and Hovy, 2011) is a link-based metric that measures coreferent and non-coreferent links. Given the set of links L_s created by the system (s), the set of links L_o created by the oracle (o) and the set of all possible links for every mention pair G, the metric calculates four scores:

1. the number of correct coreferent links - $cc = |L_s \cap L_o|$
2. the number of incorrect non-coreferent links - $in = |L_o - L_s|$
3. the number of incorrect coreferent links - $ic = |L_s - L_o|$
4. the number of correct non-coreferent links - $cn = |(G - L_s) \cap (G - L_o)|$

We then measure precision and recall as such:

$$P_c = \frac{cc}{cc + ic} \quad R_c = \frac{cc}{cc + in} \quad P_n = \frac{cn}{cn + in} \quad R_n = \frac{cn}{cn + ic}$$

$$F_c = \frac{2 \times P_c \cdot R_c}{P_c + R_c} \qquad F_n = \frac{2 \times P_n \cdot R_n}{P_n + R_n}$$

where $*_c$ refers to the metrics for coreferent links only and $*_n$ refers to the metrics for the non-coreferent links only. The overall precision, recall and F-measure are found by taking the average of the appropriate coreferent and non-coreferent scores: $P = {}^{P_c+P_n}/_2$, $R = {}^{r_c+r_n}/_2$, and $F = {}^{F_c+F_n}/_2$.

We modify the BLANC evaluation slightly to prevent undue score inflation. Our training process trains on scene-level only, so we calculate the four scores originally described by (Recasens and Hovy, 2011) for each scene. We take the sum of each type of score across all scenes and calculate precision, recall and F-measure accordingly:

$$cc = \sum_{d \in D} cc_d \quad in = \sum_{d \in D} in_d \quad ic = \sum_{d \in D} ic_d \quad cn = \sum_{d \in D} cn_d$$

where D is the set of all documents, d is a document in D and $*_d$ is one of the four scores calculated

for individual document d. Were we to calculate the confusion matrix on all mentions, correct non-coreferent links would be significantly higher, thus inflating the score.

We use the BLANC metric as a substitute for the MUC (Vilain et al., 1995) evaluation, one of the standard metrics from the CoNLL'12 shared task for coreference resolution, since both metrics are link-based metrics. However, BLANC implicitly takes into account the correctness of singleton clusters through the measurement of non-coreferent links while MUC does not, making it just as good if not better than MUC.

7 Entity Linking

7.1 Multi-Task Learning

Since coreference resolution merely clusters together similar mentions, resolving the task of coreference by itself cannot resolve the task of character identification. For the complete resolution of character identification, entity linking is necessary to assign characters (e.g., `Monica`, `Ross` in Table 2) to every mention. Thus, we mainly use coreference resolution to provide the necessary features for entity linking, which is the next step of character identification.

Figure 3 provides a visual representation of the architecture for the original entity linker. We use this model for comparison to our latest entity linking model. It can handle only singular mentions, so we make modifications to the data during the experiments to accommodate for the lack of ability to deal with plural mentions. Our latest model makes improvements upon the original entity linker, but we do not create a completely new algorithm or neural model since the main focus of entity linking in character identification rests not with the model itself but rather with the features that it is able to use. By incorporating cluster-level information, the entity linker is able to make full use of the outputs of our coreference resolution model to identify the referent characters of each mention.[7]

Figure 4 shows a visual diagram of our latest entity linking model architecture. We alter the structure of the entity linker proposed by Chen et al. (2017) to enable the model to handle both singular and plural mentions. Since we separated coreference resolution and entity linking to make character identification a two step process, the entire character identification system acts as a

[7]See Chen et al. (2017) for more details about the original approach to entity linking.

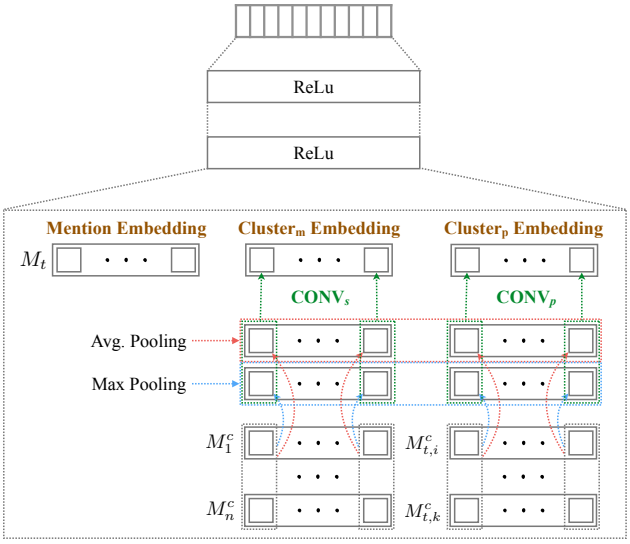

Figure 3: Original entity linking model by Chen et al. (2017).

pipeline system whereby the output of the coreference system from Section 6.1 is piped into the entity linking model. The coreference system provides the necessary raw features: the embedding of mention m_i and the set of clusters $\{C_1, \ldots, C_k\}$ to which m_i belongs. For each cluster, the system provides two types of features: mention-based cluster embedding and mention pair-based cluster embedding. We use the same techniques to generate these embeddings as Chen et al. (2017), but in order to extrapolate these techniques to plural mentions, for each plural mention m_p and its referent clusters C_p, we take the average vector of the mention-based cluster embeddings from C_p and the average vector of the mention pair-based cluster embeddings from C_p. Using the extended method, we are able to provide the entity linker with the ability to process plural mentions.

We provide two outputs as part of the joint learning process, the softmax output for singular mentions and the sigmoid output for plural mentions. Both output layers include the set of entities E; however, the softmax layer only allows the system to pick only one entity while the sigmoid layer can pick as many as it chooses. The softmax layer contains one additional label: the plural label. If the softmax layer believes that the mention is a plural mention, then it picks the plural label, which allows the system to defer to the sigmoid layer for identified characters. We use joint learning to optimize both output layers simultaneously, making this model a multi-task learning model.

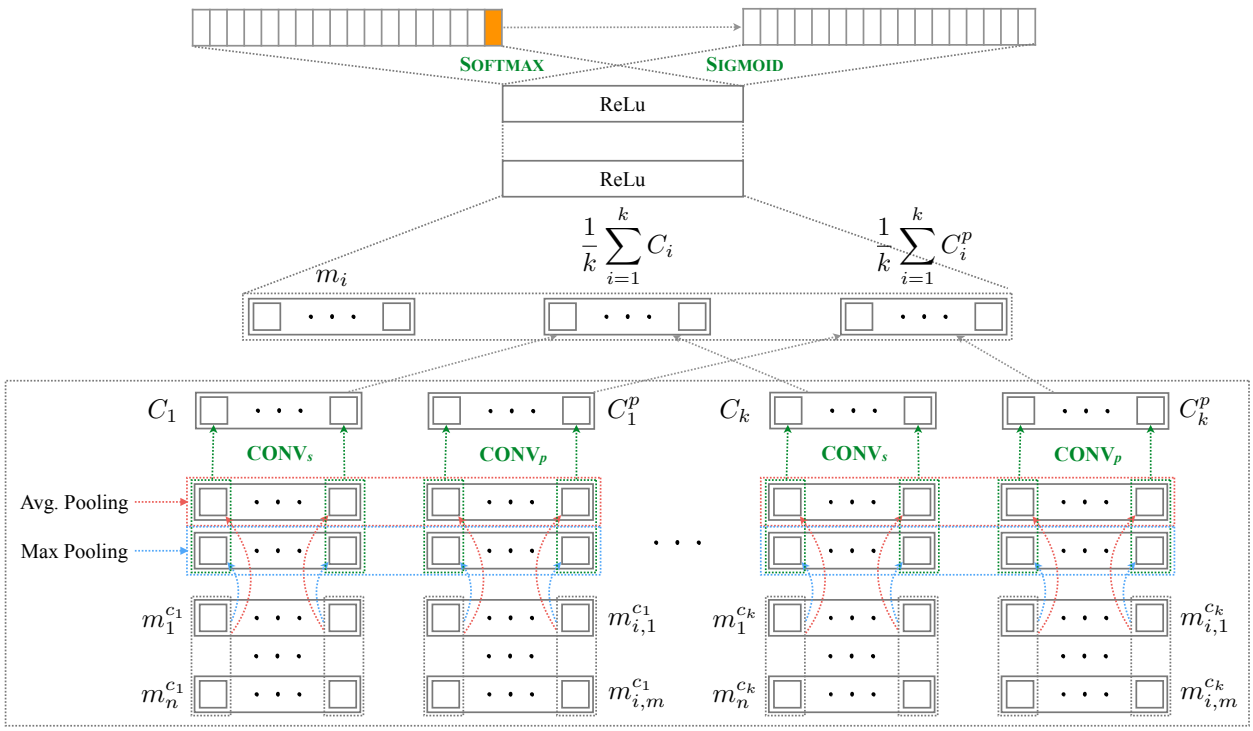

Figure 4: The overview of our entity liking model using multi-task learning.

7.2 Evaluation Metrics

We use two metrics for entity linking evaluation: micro-F_1 and macro-F_1. Micro-F_1 is a mention-based score across all documents and calculates precision (P) and recall (R) 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)):

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

Micro-F_1 essentially measures the accuracy of the system and weights frequently occurring entities more heavily. Macro-F_1 takes the micro-F_1 of every entity and takes the average: $1/|E| \sum_{e \in E} F_1^e$ where E is the set of all entities. It is more useful for gauging how well the system performs for each entity, a useful tactic when tuning the model. Of course, macro-F_1 is affected very strongly by the the frequency of an entity's appearance during training.

8 Experiments

8.1 Configuration

We conduct multiple experiments on both subtasks of character identification: coreference resolution and entity linking. Since Chen et al. (2017) use a similar two-step approach to character identification, we use models from Chen et al. (2017) as baselines for both tasks, dubbed *CZC*. These models only accept singular mentions, so we create a variation of our dataset–a pseudo-singular dataset–where every mention has exactly one referent entity, and each plural mention has the closest matching previous speaker chosen if there is one, otherwise chosen randomly. It then follows that these models may only predict one entity per mention, but are evaluated against the full dataset with singular and plural mentions. We use these models for comparison against our latest approaches described in Sections 6 and 7 which is called *Ours*. For the sake of completeness, we use the *CZC* models on the singular-only dataset variation, in which all plural mentions are filtered out, and this experiment gives a sense of how the addition of plural mentions affects model learning. All results take the average of three randomly initialized trials. We split the corpus from Section 4 into three sets: training, development and evaluation. The training set uses episodes 1–19 from each season, the development set uses epsiodes 20–21 from each season, and all remaining episodes are designated to the evaluation set. We tune all models on the development set and only the best models are tested on the evaluation set.

8.2 Coreference Resolution

Table 6 shows that our coreference model learns to handle plural mentions effectively while significantly outperforming the baseline (CZC) model. Since the CZC model is trained on the pseudo-singular dataset but evaluated on the full dataset by the modified metrics (Section 6.2), we expect that its scores will be heavily penalized for predicting only one entity for each plural mention. The results show a trend of higher precision for CZC for both B^3 and BLANC metrics, while our models show stronger performance in recall and overall F_1 score, and our model completely dominates every category in the $CEAF_{\phi_4}$ metric. This difference in performance between the

baseline and our latest approach indicates that our model effectively resolves plural mentions without incurring significant losses in its ability to identify characters for singular mentions. One can also see that the S-only model shows performance comparable to the scores by Chen et al. (2017), showing that our dataset is well constructed and that our baseline model is reliable.

	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).

8.3 Entity Linking

Tables 7 and 8 show the micro and macro average scores achieved by all models. There is a clear trend that the CZC model returns higher precision while our model achieves high recall in micro-F_1 across the board. However, the difference in precision between the baseline and the current approach is very small, suggesting that our model does not suffer from significant performance issues from attempting to resolve both singular and plural mentions. One also sees that a clear trend exists in the macro-average scores: our model dominates in all categories except for plural precision, indicating that our model has made substantial strides in learning to adequately deal with plural mentions across all entities. We expect that the micro-F_1 scores are higher than the macro-F_1 scores since the micro-average generally weights high-frequency entities more heavily while macro-average weights each entity equally. Despite posting strong gains overall, we do see some small glitches in the results–mainly that the micro-average recall for plural mentions shows relatively high variance for our model. While we did not expect high variance, we believe that running a larger number of trials would mitigate the high variance and will investigate thoroughly in the future.

Table 9 shows the micro average F1 score for each entity. We only consider the top-15 frequently appearing characters as known entities since there are hundreds of secondary characters, and most of them have negligible impact in the training data or do not appear in all three splits of the dataset. Thus, we categorize any characters not in the top-15 as `OTHER`, which composes about 26.8%. As

	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).

	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).

the results show, our model consistently outperforms the baseline on the main cast and `OTHER`, which together gives about 90% of the entire annotation. Since there is little training data for the secondary characters on the chart, we expect that there will be variation because the model does not have enough exposure to those characters. Our results are promising since they have been achieved using only system generated clusters.

	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`.

9 Conclusion

This paper demonstrates a novel guide on handling plural mentions in the two fundamental entity resolution tasks, coreference resolution and entity linking, on multiparty dialogue. We begin by addressing the inadequacy of traditional approaches in handling plural mentions and then show an innovative approach to overcome the shortcomings of these ideas for character identification. We then give a full stack overview of our approach from the expansion of the character identification

corpus along with the enhancement of the annotation framework to annotation plural mentions to the proposition of a novel coreference resolution algorithm and a deep learning-based entity linking model for a complete character identification system that comprehensively handles plural mentions. Our results show that our latest system posted strong gains on the expanded corpus, implying that our approach has promise for resolving plural mentions.

To the best of our knowledge, this paper provides the first in-depth method for resolving referents for plural mentions, which is a critical problem in coreference resolution and entity linking. We expect to explore greater improvements to our system by improving the quality of the dataset as well as expansion of its size, and addressing the issue of using global features for a more character identification system.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: Large-scale machine learning on heterogeneous systems. In *CoRR*, volume abs/1603.04467.

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.

François Chollet et al. 2015. Keras. In *GitHub repository*. GitHub.

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.

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *CoRR*.

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.