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.

Jose Coves April 8, 2019

End-to-end Plural Coreference Resolution on TV Show Transcripts

by

Jose Coves

Jinho D. Choi
Adviser

Department of Computer Science

Jinho D. Choi

Adviser

James Lu

Committee Member

Robert Roth

Committee Member

2019

End-to-end Plural Coreference Resolution on TV Show Transcripts

By

Jose Coves

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

Computer Science

2019

Abstract

End-to-end Plural Coreference Resolution on TV Show Transcripts
By Jose Coves

This paper introduces the first plural end-to-end coreference resolution model. This coreference system generates spans embeddings, which are optimized to predict the mentions and the coreferent antecedents. This model handles plural mentions and plural speakers. Our approach builds on the higher-order coreference resolution with coarse-to-fine inference by adapting it to the Friends corpus, which has plural speakers as a feature and also has singletons. Additionally, the model predicts plural antecedents as done in previous plural coreference works. These, in combination with the singular antecedents, are used to construct the final clusters, which have a one-to-one correspondence to the entities.

End-to-end Plural Coreference Resolution on TV Show Transcripts

By

Jose Coves

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

Department of Computer Science

2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Coreference resolution is a very challenging task because the model needs to understand not only the complex syntactical language structures and patterns but also a more nuanced comprehension of the meaning and nature of the dialogue, in order to accurately determine which mentions are coreferent.

Our different approaches build on this model to make it work with plural coreference resolution on the Friends corpus, which includes plural mentions, plural speakers and singletons. Singletons are mentions which are not coreferent to other mentions. They appear as a result of the way entities are characterized in the show (OTHERS, GENERAL, etc). 1.1 illustrates the differences between the annotation used in the CoNLL'12 shared task [9] and the Friends corpus, introduced by Chen and Choi [3], Zhou and Choi [11].

Document	I_0 told [mom_1 and dad_2]$_3$ last night, $they_4$ seemed to take it pretty well. Better than me_5
CoNLL'12	$\{I_0, me_5\}$, $\{mom_1\}$, $\{dad_2\}$, $\{they_4, mom$ and $dad_3\}$
Friends	$\{I_0, me_5\}$, $\{mom_1, they_4\}$, $\{dad_2, they_4\}$

Table 1.1: Annotation from CoNNL'12 and annotation from Friends [11]

In the example we can see how the new annotation by Zhou and Choi [11] does not allow

for nested mentions, like mom and dad combined. Additionally the final clustering is not a partition of the mentions, since the plural mentions appear in the clusters corresponding to the mentions they refer to. In our example $they$ is in the clusters belonging to mom and dad respectively, since those are the entities it is referencing. Plural mentions might not refer to the exact same group of entities but still have some in common. However, instead of making plural mentions coreferent, it makes more sense to link them to each singular mention and the corresponding entity it represents. Coreference resolution is a very powerful and essential task in Natural Language Processing, especially when combined with other tasks. Higher-level tasks like machine translation, text comprehension and question answering can greatly benefit from coreference resolution, since it helps identify the entities the mentions refer to.

Plural coreference resolution is a relatively new task that has not been explored in depth. It was introduced recently by Zhou and Choi [11] and their approach consisted of a modified version of an Agglomerative Convolutional Neural Network to classify every pair of mentions as either non-coreferent, left-coreferent or right-coreferent. These labels were used to then construct the clusters of mentions, where clusters and entities had a one-to-one correspondence. While there is previous work on both end-to-end neural coreference resolution and plural coreference resolution, we believe this is the first time end-to-end plural coreference resolution has been done.

Chapter 2

Background

Coreference resolution is a task that belongs to the Natural Language Processing (NLP) field. This field attempts to model and learn language structures and patterns by using programming methods and algorithms. In the earlier stages of NLP, the algorithms used for most tasks relied more heavily on domain knowledge and many hand-crafted rules and heuristics from linguistic experts.

Coreference resolution is the task of partitioning all expressions into the entities they refer to. This expressions are often called mentions and include, but are not limited to, names, nouns and pronouns: e.g. *Joey, guy, he*. The objective is to group up the mentions referring to the same entity into the same cluster, yielding a one-to-one correspondence between clusters and entities. Entities could be as general as things, locations, people or objects, like in the CoNLL'12 corpus. However, other corpora may have different or more specific definitions of entities. The Friends corpus only considers the characters in the show, thus limiting the entities to people. Traditionally, coreference resolution only involved singular mentions, which only refer to one entity and therefore only belong to one cluster.

Thus, the cluster {*they, mom and dad*}, as annotated in CoNLL'12 is technically a cluster formed by two singular mentions, despite the fact that both mentions refer to a plural entity, which is an entity that refers to multiple entities. The annotation in the friends corpus addresses this by removing the nested mention *mom and dad* and annotating $they$ as a plural mention, which is a mention that belongs to multiple cluster since it refers to multiple entities. As a result, it is possible to deal with mentions that refer to multiple entities, such as $they$, while still having the property of the clusters corresponding to entities (*mom and dad* is not an entity, but rather two separate entities). This is very important since it allows the work to be expanded to perform Entity Linking, a task in which the coreferent clusters are paired with the entities they refer to.

As neural networks became more popular due to impressive boosts in performance, the NLP field also started to be dominated by state-of-the-art deep learning models. One of the biggest challenges was that words do not have a numerical representation, and therefore could only be encoded as sparse vectors. Nowadays, however, there exist many efficient libraries for learning word representations, even at character-level, which makes it possible to obtain numerical representations for previously unseen words.

This paper expands on the end-to-end neural networks built using the popular libray TensorFlow [1]. The library supports a variety of computational methods. For a better understanding of the model, we explain the two main neural networks used:

- Feedforward Neural Network (FFNN) - It is a multi-layer perceptron, which feeds its output forward as an input to the next layer, without forming a cycle.

- Long Short-Term Memory (LSTM) - It is an artificial Recurrent Neural Network (RNN), i.e., the connections between the nodes form a directed graph with loops which allow information to persist. It has different gates that determine when to forget and remember new information. The main advantage is that it can process entire sequences of data, such as speech.

- Convolutional Neural Network (CNN) - It is a class of deep neural networks that extracts the most significant and salient features from the input in a condensed form. They have been mostly used for extracting salient features from visual data and images but have also been proved useful in NLP for processing sentences and words.

2.1 Related Work

There is substantial related work done on coreference resolution, but we will focus on explaining the most relevant to understand the work we present in this paper.

Many of the original coreference models relied heavily on syntactic parsers and featured rule based hand-engineered systems for mention detection. Clark and Manning [4] optimized a neural mention-ranking model for coreference by applying reinforcement learning. Clark and Manning [5] presented a learning-to-search algorithm that optimized

local decisions for merging the clusters. Wiseman et al. [10] used recurrent neural networks to learn representations for the entitiy clusters from their mentions. More recently we saw the appearance of state-of-the-art coreference models that are end-to-end [7, 8] and make no assumptions about the syntax or structure of the language in order to identify which spans are most likely mentions. In a very efficient manner, these span embeddings used to obtain the mention scores are then refined and used to predict the antecedents.

End-to-end coreference

End-to-end neural coreference resolution [7, 8] produces the coreferent clusters by assigning antecedents to the top spans. For each span i, the set of antecedents $Y(i) = \{\epsilon, 1, ..., i-1\}$, where ϵ is the dummy antecedent, used when span i is not a mention or not coreferent.

The first step is to compute the span embeddings and mention scores, which is done using a bidirectional LSTM, as shown in 2.1.

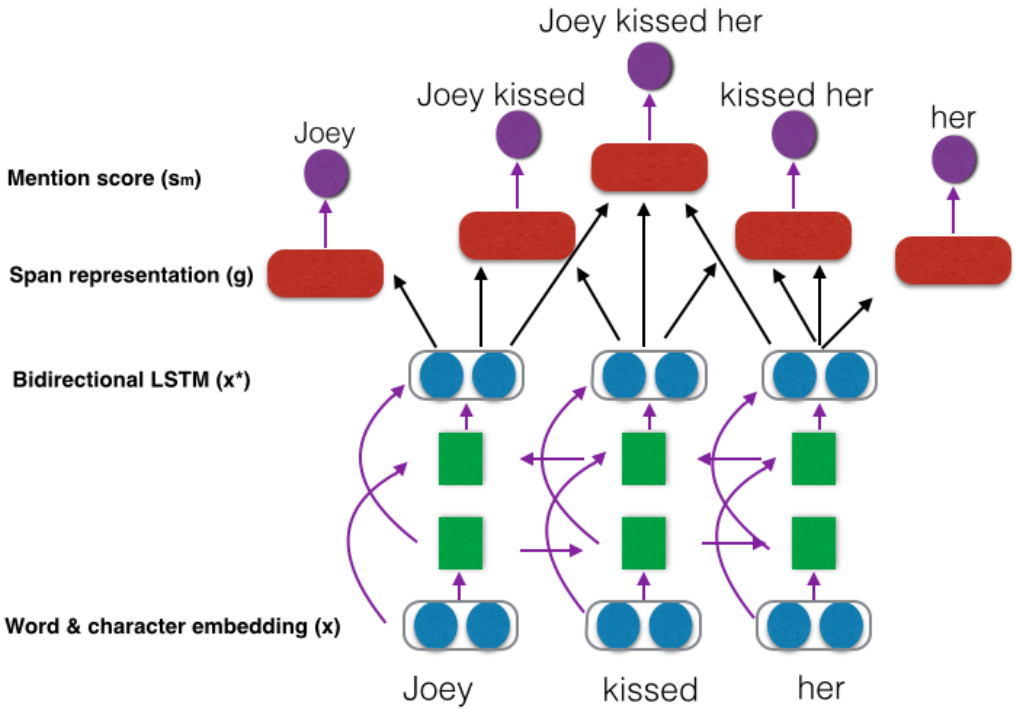
$$S(i,j) = s_m(i) + s_m(j) + s_a(i,j) \tag{2.1}$$

$$S_m(i) = W_m^T \text{FFNN}(g_i) \tag{2.2}$$

$$S_a(i,j) = W_a^T \text{FFNN}([g_i, g_j, g_i \circ g_j, \phi(i,j)]) \tag{2.3}$$

where g_i is the span representation of i, $\circ$ denotes element-wise multiplication and $\phi(i,j)$ is a feature vector containing speaker information and a distance factor between the mentions.

Figure 2.1: Feed character and word embeddings into bidirectional LSTM to compute the span representations and mention scores for potential entities.

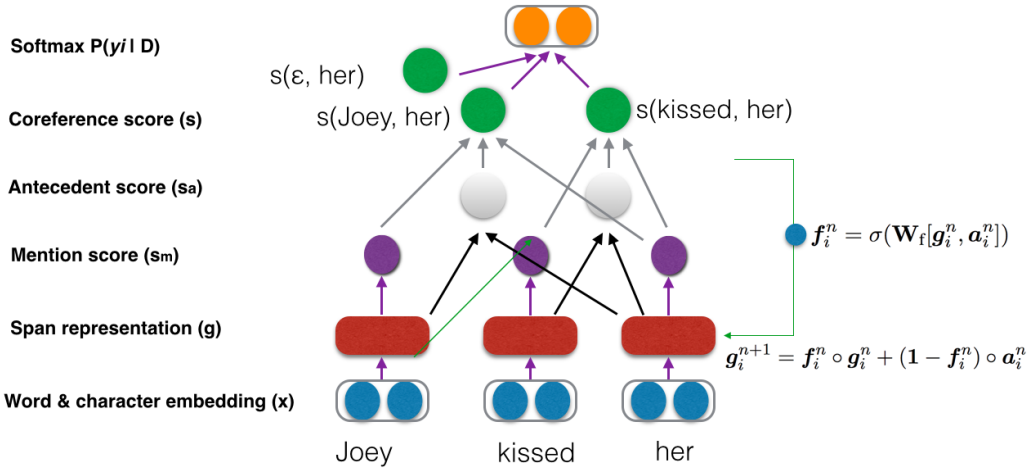


W_m, W_a and W_c (2.5) are learnt weight matrices.

The next step in the end-to-end model is to feed the span representations into the Feedforward Neural Networks to compute the mention scores and the antecedent scores, which are then combined to obtain the coreference score, as displayed in figure 2.2. The coreference score $S(i, j)$ (2.1) of a pair of mentions is obtained by adding their mention scores (2.2) and their antecedent scores (2.3). For each span, the antecedent with the highest coreference score is selected. If the span is not a mention or is not coreferent, the dummy

antecedent ϵ is selected.

Figure 2.2: End-to-end coreference system. The span representations are simultaneously used for computing the mention scores and the antecedent scores. These are combined to obtain the final coreference scores.



Heavy pruning occurs by sorting the spans based on the mention scores and only keeping the top $K = \lambda T$ spans, where λ is the span ratio and T is the length of the document.

In their following paper, Lee et al. [8] add higher-order coreference resolution, which is achieved by refining the span representations for N iterations. At each iteration, the expected antecedent representation a_i^n, in combination with the gate vector f_i^n, is used to modify the current span representation g_i^n (2.4).

$$g_i^{n+1} = f_i^n \circ g_i^n + (1 - f_i^n) \circ a_i^n \tag{2.4}$$

The other big addition from the second paper is coarse-to-fine inference, which consists of pruning antecedents using a bi-linear matrix, making it less accurate but much faster.

This increase in efficiency allows the model to, for each span in the top K, select the top C antecedents, ranked using (2.5). Then, the final coreference score for these top antecedents is computed using (2.6). This equation is very similar to the original, but it now has the extra term for the fast antecedent score.

$$s_c(i,j) = \mathbf{g}_i^T \mathbf{W}_c \mathbf{g}_j \tag{2.5}$$

$$S(i,j) = s_m(i) + s_m(j) + s_a(i,j) + s_c(i,j) \tag{2.6}$$

Plural coreference

Plural coreference resolution, which involves plural mentions, was introduced by Zhou and Choi [11] only very recently, given the complexity of the task and the lack of corpora with annotation for plural mentions. The Friends corpus has been revised and annotated to support plural coreference resolution. Previously, the corpus had been used for singular coreference resolution and entity linking [2]. Their model consisted of an Agglomerative Convolutional Neural Network (ACNN), which aggregates convolutions layers to learn the representations for the mentions and then use these to compute the mention pair embeddings.

One of the main challenges of plural coreference is that it is no longer transitive. Singular coreference resolution can be seen as a partition of the mention into clusters, where each cluster represents an entity. However, the introduction of plural mentions,

which refer to multiple entities and therefore appear in multiple clusters, means that we must now broaden our search space.

For singular coreference resolution, the algorithm by Lee et al. [7] just finds for every mention i an antecedent mention j and combines them into the same cluster in a manner similar to a disjoint set union find, given the transitive nature of coreference. However, this does not hold true for plural coreference resolution, since we can have the clusters $\{me, we\}$, $\{you, we\}$ where we is coreferent to both mentions but you and me are not coreferent. Therefore, plural coreference resolution involves comparing all pairs of mentions. Additionally, now there must be two type of links between pairs of mentions, to differentiate between singular and plural antecedents. In their original paper they are referred to as L and R, respectively. The following algorithm is described for clustering is performed:

- Singular antecedent: when m_i is singular. If m_i is not in a cluster, create and assign it. m_j gets assigned to the cluster m_i belongs to.
- Plural antecedent: when m_i is plural and m_j is singular. If m_j is not in a cluster, create and assign it. m_i gets assigned to the cluster m_j belongs to.
- Not coreferent: every other case. Do nothing.

The mention pairs are iterated in their natural order of appearance, as shown in 2.1

$[\mathbf{m_i}] \rightarrow \{\mathbf{N}, \mathbf{S}, \mathbf{P}\}$	$\mathbf{m_j}$	**Clusters**
$[I] \rightarrow N$	mom	$\{\}$
$[I, mom] \rightarrow N$	dad	$\{\}$
$[I] \rightarrow N, [mom, dad] \rightarrow S$	$they$	$\{mom, they\}, \{dad, they\}$
$[I] \rightarrow S, [mom, dad, they] \rightarrow N$	me	$\{mom, they\}, \{dad, they\}$, $\{I, me\}$

Table 2.1: Clustering algorithm. N,S,P stand for non-coreferent, singular antecedent, and plural antecedent, respectively

Since the task of plural coreference resolution has different gold labels from singular coreference, the evaluation metrics were revisited and modified by Zhou and Choi [11]. We use those updated evaluation metrics, Bcube, Ceafe and Blanc.

Chapter 3

Approach

The main novelty compared to previous work on Coreference resolution is that now the model deals with plural mentions. Previous work on plural Coreference is very limited and was explored on the Friends corpus. This paper expands on the previous work by adding an extra layer of complexity, in that the model is not given the gold mentions. Instead, it has to not only predict the clusters the mentions belong to, but also the mentions themselves, thus achieving end-to-end plural coreference resolution.

As inspiration to tackle this problem, the ideas and algorithms from the top performing models in end-to-end coreference resolution [8] and plural coreference [11] are combined. This merge itself presents many difficulties, many of which arise because they are designed to deal with different corpora. The main difference is that the end-to-end model uses CoNLL corpus, which consists of singular mentions, includes nested mentions and has no singletons. On the other hand, the Friends corpus has both plural and singular mentions, none of which are nested, and includes some singletons due to having clusters for general or unnamed characters. Many of these differences are a direct consequence of the fact

that the Friends corpus only consists of mentions referring to people, whereas CoNLLs mentions include people, locations, organizations and objects. Additionally, since 99% of the mentions are at most three words long, the maximum span width is set to that value.

3.1 Nested Mention Detection

After learning about the successful results of end-to-end neural coreference resolution we were a bit skeptical about the effectiveness of mention detection. Therefore, we aimed to improve the performance by adding the state-of-the-art mention detection model presented by Katiyar and Cardie [6]. However, the recurrent neural network hypergraph-based model did not improve the recall of mention detection. This further supports the effectiveness and strength of the end-to-end system. Thus, we focus all of our following approaches on expanding it to plural coreference resolution.

3.2 Plural Speakers

The baseline for our model consists of the e2e model designed for singular Coreference resolution on the CoNLL corpus. In order to adapt it to perform on the Friends corpus, the speakers feature is expanded to include plural speakers. The algorithm previously used the feature as a binary flag on whether two mentions had the same speaker. It is modified in this plural version so it represents whether two mentions have any speakers in common,

i.e., their intersection was non-empty. To allow an easy comparison between singular and plural speakers, the results are split in two separate tables. Additionally, we include the results for a different use of the speakers feature in 4.7 and 4.8. Instead of having a feature vector to denote whether both mentions have the same speaker, now the model assigns random embeddings to all the different speakers (around 300). Then, the embeddings for both speakers' mentions, as well as their pair-wise similarity (computed as element-wise product) is appended to the feature vector.

The advantage from this modification can be seen in the following example:

John: I won.

Mary: ME too.

John,Mary: WE rock!

Technically, all three utterances have different sets of speakers, but it is clear that there is significance in having a non-empty intersection of speakers. Now both (I, WE) and (ME, WE) will have the same speaker embedding, which can be very helpful for finding coreference between first person pronouns.

3.3 Singletons

Since the CoNLL corpus does not have singletons but the Friend corpus does, mention scores were used to find such singletons. The process consists of adding mentions that had

not been assigned to any clusters but had a mention score over a threshold t. This parameter can be tuned and the optimal value was found to be approximately $t = 0$, which makes sense since Lee et al. [7] determined that while the initial pruning is completely random, only gold mentions receive positive updates. This process of adding remaining mentions (not predicted as coreferent to other mentions) is performed on every approach.

3.4 Training Labels

Finally, to get a baseline for plural coreference the gold labels for plural mentions (for training only) had to be modified. This baseline would only output singular clusters, in a similar manner to the baseline used by Zhou and Choi [11]. Every plural mention is formed by many singular mentions, so one of these mentions, the ”head mention”, had to be selected in order to convert the training labels into singular labels. This would allow for evaluating a singular coreference model on a plural corpus with plural metrics. Different heuristics were explored to select the head mention. These involved sorting by the frequency of appearances of the mentions in the document and picking either the most or the least popular. These correspond to *Singular + most* and *Singular + least* in the results table. It was also considered ignoring plural mentions during training (*Singular + none* in the results).

3.5 Plural Coreference Resolution

Once the model has been adapted to work on the Friends corpus, more modifications are made to actually address plural mentions. For singular coreference resolution, the algorithm just finds for every mention i an antecedent mention j and combines them into the same cluster in a manner similar to a disjoint set union find, given the transitive nature of coreference. However, this does not hold true for plural coreference resolution, since it is possible to have the clusters $\{me, we\}$, $\{you, we\}$ where we is coreferent to both mentions but you and me are not coreferent. Therefore, plural coreference resolution involves comparing all pairs of mentions. Additionally, now there must be two type of links between pairs of mentions, to differentiate between singular and plural antecedents.

$$S_{pa}(i,j) = W_{pa}^T \text{FFNN}([g_i, g_j, g_i \circ g_j, \phi(i,j)]) \tag{3.1}$$

$$s_{pc}(i,j) = \mathbf{g}_i^T \mathbf{W}_{pc} \mathbf{g}_j \tag{3.2}$$

$$S_p(i,j) = s_m(i) + s_m(j) + s_p(i,j) + s_{pc}(i,j) \tag{3.3}$$

$$Loss = S * Loss_{Singular} + (1 - S) * Loss_{Plural} \tag{3.4}$$

3.6 Singularity

In order to model this additional relationship between pairs of mentions, S_{pa}, score of plural antecedents, a separate feed forward neural network is added, identical to the one used for singular antecedents. Equations (3.1-3) show the parallelism with regards to how the singular antecedents scores are computed. Thus, the model arrives at coreferent scores for both singular and plural antecedents. The difference is that now the loss is computed as a weighted average of the singular and plural antecedents losses. The weights are determined by the parameter singularity S which at a value of 1 only looks at the singular antecedents loss. Since singular antecedent relationships are more common than plural ones, they are more important, and the optimal values of S are between 0.6 and 0.7.

This also affects higher-order coreference resolution and the refining of the span representations. Now, at each iteration, the expected singular and plural antecedents representations, a_i^n and pa_i^n, in combination with the gate vector f_i^n, are used to modify the current span representation g_i^n (3.1).

$$g_i^{n+1} = f_i^n \circ g_i^n + (1 - f_i^n) \circ (Sa_i^n + (1 - S)pa_i^n) \tag{3.5}$$

The first version of the plural clusters model is trained to predict a singular antecedent i for a span j if they are coreferent (i.e., they belong to the same gold cluster) and the span j is a singular mention, regardless of whether the antecedent i is singular or plural. Whether a

mention is plural or singular is determined by looking at how many gold clusters it belongs to, since singular mentions only belong to one cluster. Plural antecedents are to be predicted when the antecedent i is a singular mention and the span j is plural. As argued by Zhou and Choi [11], it is desirable to avoid having clusters just with plural mentions, since the idea of coreference resolution in dialogue is to match mentions and clusters with entities. Therefore, two plural mentions are not predicted as plural antecedents. The labeling for the utterance "I told $mom\ they$ need me. Dad agreed." is shown in 3.1

Antecedent $\mathbf{m_i}$	**Span $\mathbf{m_j}$**	**Antecedent Label**
I	me	Singular
mom	$they$	Plural
$they$	dad	Singular

Table 3.1: Labeling algorithm. Non-coreferent labels are not included.

3.7 Merging Clusters

The first approach to produce the clusters uses the baseline method to compute the clusters using only singular antecedents, which adds m_i to the cluster m_j belongs to if (m_i, m_j) are coreferent . Then the plural antecedents are used to expand the existing clusters. This is done by adding the plural span m_j to the cluster the antecedent mention m_i belongs to (if it has no cluster it creates a new one). The results of this approach are displayed in the table as the experiment *Base + plural*. The main weakness of this approach is that it still assumes the transitivity of singular coreference resolution when constructing the clusters

from the singular antecedents and it only looks at the top antecedent for each span, instead of looking at all pairs.

Example: I *think* we *won.* You *did great.*

The pair (we, you) forms a singular antecedent, so we is added to the cluster you belongs to. On the other hand, (I, we) is a plural antecedent, so we is added to the cluster I belongs to, resulting in the clusters $\{$I, we$\}$ and $\{$you, we$\}$.

3.8 Many Antecedents with Upper Bounds

The next approach attempts to solve the previous weakness by looking at many singular antecedents. Previously, only the top singular antecedent was selected, unless it was the dummy antecedent ϵ. This happens because the original end-to-end system picks the antecedent using Softmax regression, as shown in 2.2. This issue is addressed by applying the same algorithm used for plural antecedents: for a given span, all antecedents whose score is greater than the dummy antecedent ϵ are selected. This differs from Zhou and Choi [11], where the model is a classifier and it already produces the labels. In order to avoid too many antecedents being selected, the model includes two parameters *max singular antecedents* and *max plural antecedents* which provide an upper bound on the number of antecedents. Thus, in the sentence "I bought it for me, but we could share it." the antecedent scores of I and me for span we are both greater than the dummy ϵ and they

are both marked as singular antecedents, and not just the one with the highest scores. The results are shown in *Plural + many*.

3.9 Antecedent conflicts

The concern with the previous method is that it does not follow the natural order of the pairs of mentions. Instead, all the singular antecedents are processed first and only then the plural antecedents are used to combine the clusters. This is not a problem if the predicted labels are all correct, but in practice there are wrong predictions and their negative effect can be reduced. Hence, the next improvement looks at all spans in order of appearance, and for each span it looks at all antecedents in order. This also deals with the issue of having a pair of mentions marked as both singular and plural antecedents. If this happens, the model selects whichever has the highest score. These results are shown as *Many + order* in the results. Consider the example: I *think* we *won.* You *did great.*

The correct labels are $(I, we) \rightarrow$ *Singular Antecedent* and $(we, you) \rightarrow$ *Plural Antecedent*. If we also add the incorrect prediction $(I, you) \rightarrow$ *Singular Antecedent* we get the following clusters. After this modification we improved the prediction from both clusters being wrong 3.2 to only one wrong cluster 3.3. This is because errors are propagated, and spans that appear later have more possible antecedents and more predictions that could be wrong, Thus, by processing their antecedents at a later stage, any mistakes are not as influential.

Antecedent, Span	Label	Clusters
I, we	Singular	$\{I, we\}$
I, you	Singular	$\{I, we, you\}$
we, you	Plural	$\{I, we, you\}$

Table 3.2: Clustering obtained by processing singular antecedents first. Utterance has wrong prediction $(I, you) \rightarrow$ *Singular Antecedent*

Antecedent, Span	Label	Clusters
I, we	Singular	$\{I, we\}$
we, you	Plural	$\{I, we\}, \{you, we\}$
I, you	Singular	$\{I, we, you\}, \{you, we\}$

Table 3.3: Clustering obtained by sorting the processing order by (span, antecedent). Utterance has wrong prediction $(I, you) \rightarrow$ *Singular Antecedent*

3.10 Alternate plural antecedents

After trying different approaches for constructing the clusters, the next iteration of the model defines a new training loss for the plural antecedents. Until now the mention pair (i,j) was a plural antecedent if the mention span j is plural and the antecedent i is singular. However, now the model takes an approach similar to that of Zhou and Choi [11] and the mention pair (i,j) is a plural antecedent if the mention span j is singular and the antecedent i is plural. Of course, this requires to modify the algorithm for construction of the clusters. In fact, this now requires the clustering algorithm described in Zhou and Choi [11]. The clustering for the utterance "I told $mom\ they$ need me. Dad agreed." is shown in 3.4. The results are shown as *New Plural* in the experiments.

In the above approach, the cluster construction is still done in two phases: first, just

$[\mathbf{m_i}] \rightarrow \{\mathbf{N}, \mathbf{S}, \mathbf{P}\}$	$\mathbf{m_j}$	**Clusters**
$[I] \rightarrow N$	mom	$\{\}$
$[I] \rightarrow N, [mom] \rightarrow S$	$they$	$\{mom, they\}$
$[I, mom] \rightarrow N, [they] \rightarrow P$	dad	$\{mom, they\}, \{dad, they\}$
$[I] \rightarrow S, [mom, dad, they] \rightarrow N$	me	$\{mom, they\}, \{dad, they\}$, $\{I, me\}$

Table 3.4: Clustering for the utterance "I told mom $they$ need me. Dad agreed.". N,S,P stand for non-coreferent, singular antecedent, and plural antecedent, respectively

looking at the singular antecedents and obtaining the partial clusters, which are then used in combination with the plural antecedents to construct the final clusters. Thus, in this new experiment the same predictions obtained in ln are used but now the final clusters are constructed using singular and plural antecedents simultaneously as the spans are iterated in order of appearance.. This results are shows as *New Plural + order*.

Chapter 4

Experiments

The Friends corpus is divided into training, development and test in the following manner. Episodes 1-19 are for training, 20-22 for development, and the remaining episodes are for the test set. The results are presented in two different tables, corresponding to singular and plural speakers. The most important baseline is the state-of-the-art model for plural coreference resolution introduced by Zhou and Choi (ZC) [11], which was covered in depth in 2.1. Additionally, we include the model by Chen, Zhou and Choi (CZC) for reference. CZC is a singular coreference resolution model by Chen et al. [2] which was modified to use plural labels for training. Therefore, CZC predicts singular clusters and it served as a baseline for the more advanced model developed for ZC. Both ZC and CZC, covered in 2.1, use gold mentions during evaluation, so our end-to-end model has an extra layer of complexity since it needs to detect the mentions.

By comparing the results from both tables, it is clear that plural speakers (table 4.6) have a slight boost in performance over singular speakers (table 4.3), since the scores are consistently higher. This confirms our hypothesis that the emptiness of the intersection

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Singular + none	95.3	83.3	88.9	74.4	66.3	70.1	**62.5**	44.4	52.0	75.7	70.5	72.2	65.8
Singular + least	96.0	90.2	93.0	70.0	64.5	67.2	60.8	44.5	51.4	75.0	73.1	73.9	65.2
Singular + most	95.7	90.3	92.9	72.6	65.5	68.8	60.4	44.8	51.4	77.4	75.0	76.0	66.3
Base + plural	93.5	96.2	94.8	72.7	60.0	65.8	44.5	57.4	50.1	77.0	73.2	74.8	65.1

Table 4.1: Results on the test set with singular speakers

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Base + plural	93.5	96.2	94.8	72.7	60.0	65.8	44.5	57.4	50.1	77.0	73.2	74.8	65.1
Plural + many	95.8	88.7	92.2	69.3	70.6	70.0	52.2	48.0	50.0	74.3	75.9	75.0	64.4
Many + order	**96.2**	90.5	93.3	70.0	66.9	68.4	59.5	45.9	51.8	75.7	74.6	75.1	65.6
New Plural	94.8	93.2	94.0	67.3	72.5	69.8	56.3	48.6	52.2	75.0	78.1	76.3	66.6
N. Plural + order	95.0	**95.0**	**95.0**	67.9	**72.7**	70.1	52.5	55.3	53.9	76.3	**79.2**	77.6	67.0

Table 4.2: Results on the test set with singular speakers

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Singular + none	95.3	83.3	88.9	74.4	66.3	70.1	**62.5**	44.4	52.0	75.7	70.5	72.2	65.8
Singular + least	96.0	90.2	93.0	70.0	64.5	67.2	60.8	44.5	51.4	75.0	73.1	73.9	65.2
Singular + most	95.7	90.3	92.9	72.6	65.5	68.8	60.4	44.8	51.4	77.4	75.0	76.0	66.3
Base + plural	93.5	96.2	94.8	72.7	60.0	65.8	44.5	57.4	50.1	77.0	73.2	74.8	65.1
Plural + many	95.8	88.7	92.2	69.3	70.6	70.0	52.2	48.0	50.0	74.3	75.9	75.0	64.4
Many + order	**96.2**	90.5	93.3	70.0	66.9	68.4	59.5	45.9	51.8	75.7	74.6	75.1	65.6
New Plural	94.8	93.2	94.0	67.3	72.5	69.8	56.3	48.6	52.2	75.0	78.1	76.3	66.6
N. Plural + order	95.0	**95.0**	**95.0**	67.9	**72.7**	70.1	52.5	55.3	53.9	76.3	**79.2**	77.6	67.0
CZC	-	-	-	84.5	60.7	70.6	49.0	63.7	55.4	**81.2**	73.3	75.9	67.7
ZC	-	-	-	**83.8**	67.0	**74.4**	52.1	**68.0**	**59.0**	80.4	76.5	**78.0**	**70.6**

Table 4.3: Results on the test set with singular speakers

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Singular + none	94.3	86.8	90.4	74.4	65.5	69.6	54.6	52.5	53.5	76.4	71.3	73.1	65.2
Singular + least	95.5	90.4	92.9	72.8	62.2	67.1	59.4	47.5	52.9	77.2	73.2	74.8	64.3
Singular + most	95.7	90.6	93.1	73.7	65.2	69.1	**60.1**	46.0	52.2	79.3	75.4	77.0	65.6
Base + plural	95.1	88.4	91.7	72.8	64.6	68.5	57.4	45.7	50.9	77.0	74.0	75.3	63.8

Table 4.4: Results on the test set with plural speakers

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Base + plural	95.1	88.4	91.7	72.8	64.6	68.5	57.4	45.7	50.9	77.0	74.0	75.3	63.8
Plural + many	96.1	93.1	94.6	66.4	70.3	68.3	50.1	51.4	50.8	72.8	74.9	73.8	65.2
Many + order	**96.3**	89.0	92.5	74.0	64.9	69.1	56.4	47.4	51.5	77.6	74.1	75.6	65.2
New Plural	95.1	**95.1**	**95.1**	66.7	72.6	69.5	52.8	53.7	53.3	74.9	78.3	76.3	66.3
N. Plural + order	96.1	91.3	93.7	69.1	**72.7**	70.8	56.8	48.1	52.2	77.0	**78.5**	77.7	67.4

Table 4.5: Results on the test set with plural speakers

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Singular + none	94.3	86.8	90.4	74.4	65.5	69.6	54.6	52.5	53.5	76.4	71.3	73.1	65.2
Singular + least	95.5	90.4	92.9	72.8	62.2	67.1	59.4	47.5	52.9	77.2	73.2	74.8	64.3
Singular + most	95.7	90.6	93.1	73.7	65.2	69.1	**60.1**	46.0	52.2	79.3	75.4	77.0	65.6
Base + plural	95.1	88.4	91.7	72.8	64.6	68.5	57.4	45.7	50.9	77.0	74.0	75.3	63.8
Plural + many	96.1	93.1	94.6	66.4	70.3	68.3	50.1	51.4	50.8	72.8	74.9	73.8	65.2
Many + order	**96.3**	89.0	92.5	74.0	64.9	69.1	56.4	47.4	51.5	77.6	74.1	75.6	65.2
New Plural	95.1	**95.1**	**95.1**	66.7	72.6	69.5	52.8	53.7	53.3	74.9	78.3	76.3	66.3
N. Plural + order	96.1	91.3	93.7	69.1	**72.7**	70.8	56.8	48.1	52.2	77.0	**78.5**	77.7	67.4
CZC	-	-	-	84.5	60.7	70.6	49.0	63.7	55.4	**81.2**	73.3	75.9	67.7
ZC	-	-	-	**83.8**	67.0	**74.4**	52.1	**68.0**	**59.0**	80.4	76.5	**78.0**	**70.6**

Table 4.6: Results on the test set with plural speakers

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Singular	95.1	**95.1**	**95.1**	66.7	72.6	69.5	52.8	**53.7**	**53.3**	74.9	**78.3**	**76.3**	**66.6**
Average	94.8	93.2	94.0	**67.3**	72.5	**69.8**	**56.3**	48.6	52.2	**75.0**	78.1	**76.3**	65.9
Plural	**95.8**	91.9	93.8	66.4	**73.3**	69.6	55.2	47.9	51.3	74.7	77.9	76.0	66.3

Table 4.7: Results on the test set for *New Plural* with different speakers feature.

	Mention			Bcube			Ceafe			Blanc			Avg
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Singular	95.0	**95.0**	**95.0**	67.9	**72.7**	70.1	52.5	**55.3**	**53.9**	76.3	**79.2**	77.6	67.0
Average	95.1	93.0	94.1	66.5	71.8	69.1	50.6	52.1	51.4	74.8	77.6	76.0	65.7
Plural	**96.1**	91.3	93.7	**69.1**	**72.7**	**70.8**	**56.8**	48.1	52.2	**77.0**	78.5	**77.7**	**67.4**

Table 4.8: Results on the test set for *New Plural + order* with different speakers feature.

of a set of speakers can help determine the coreference of mentions. Given the superior performance of using plural speakers, the rest of the section analyzes the results for plural speakers. Tables 4.7 and 4.8 show the differences in performance as a results of using the proposed approaches for the speakers features. Overall, averaging the speakers' representations does not yield very promising results. This is most likely a consequence of the much larger vector space the model needs to consider, since now there are over 300 randomly initialized speaker vectors, whereas before it was only the binary *same_speaker* vector. A future improvement would be to use less speaker vectors (maybe only for the top 50 speakers) and actually train their representations.

Looking at the models that predict singular clusters, it is not surprising that ignoring plural clusters would result in a bad performance, since many labels are not used during the training, resulting in less opportunities for the model to learn. Using the most popular entity yields better results and it makes sense intuitively, since the most popular entity has the most mentions, and therefore getting its mentions in the wrong cluster carries a larger penalty. It is noticeable how this model outperforms every other approach introduced in this paper except for $NewPlural$ and $NewPlural + order$, reminding of the complexity and difficulty of predicting plural antecedents and dealing with prediction noise during the construction of the clusters.

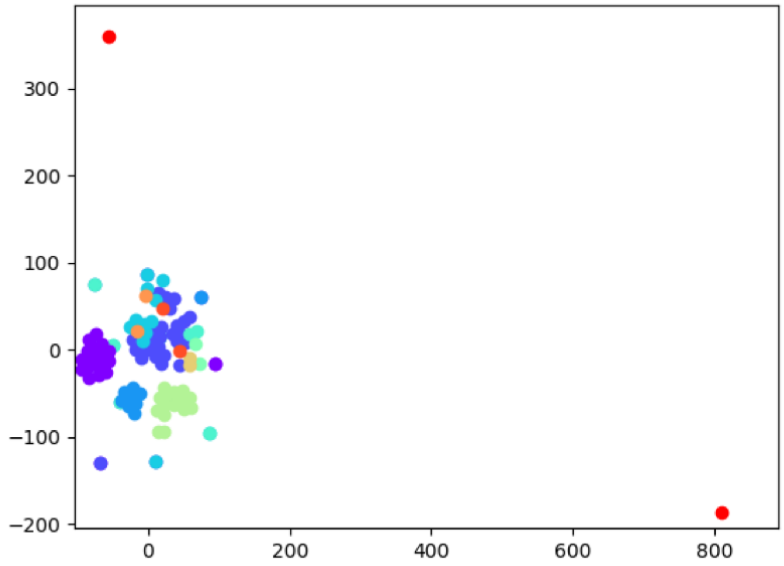
The first plural coreference resolution approach, $Base + plural$, successfully predicts

clusters with plural mentions, but the predictions still have errors which accumulate and hurt the performance. After addressing the issue of only predicting at most one singular antecedent per span, $Plural + many$ actually performs worse, given that its side-effect was to augment the previous issue of error accumulation. This is finally fixed in $Many + order$, whose performance starts to get closer to $Singular + most$.

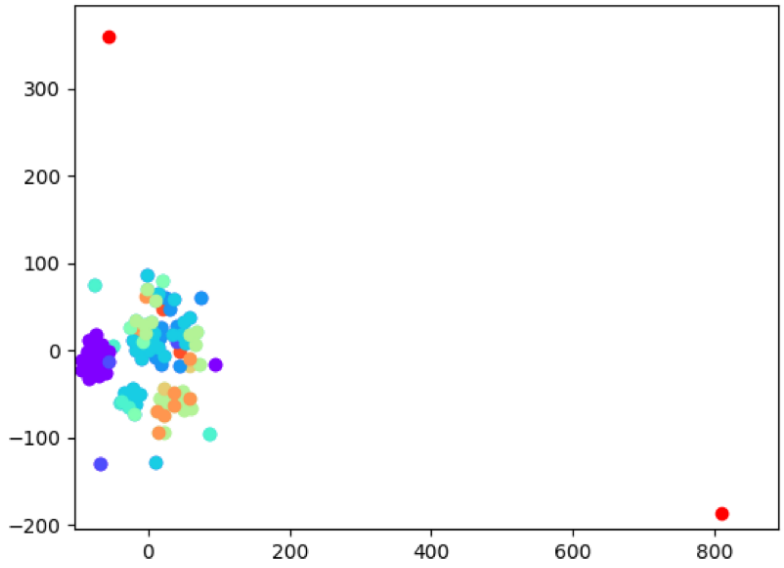
$NewPlural$ introduces a new labeling for plural and singular antecedents and it outperforms all previous approaches. Once the accumulation error is mitigated in $NewPlural + order$, the results outperform every previous approach and match evenly with CZC (lower Ceafe but higher Blanc score). ZC still outperforms this approach but given the fact that the end-to-end model does not use gold mentions on the test set, its very close Blanc score is significant. $NewPlural + order$ achieves the best mention detection, with an F1 score of 95.1.

Figures 4.1 and 4.2 show the clusters each of the mentions were assigned to. Each color corresponds to a cluster, and therefore all mentions labeled with the same color refer to the same entity. However, the clusters have not yet been matched with their respective entities, which would happen during the entity linking task. The location of the mentions in the figures are the representations of the mentions in the two-dimensional space, after applying dimension reduction in order to visualize it. The improved approach *New Plural* has a more spread out mention representation, which makes it more meaningful when it comes to predicting the antecedents. Additionally, by comparing both the predicted clusters with the gold clusters, it is noticeable how the model struggles by splitting two clusters, when they should be merged into one cluster. For example, in 4.3(b) we have the orange cluster around $x = 6$ but the model splits them into two clusters (orange and cyan) in 4.3(a). Another key observation revealed by the gold clusters is that the clusters cannot be explained solely by the mention embeddings, which suggests that coreference resolution is a very hard task that requires a complex model capable of successfully processing syntactical structures, sequential references and metadata such as speaker information.

Figure 4.1: Mention representation for Base + Plural. The axes represent the mentions' embeddings after applying dimension reduction. All mentions sharing the same color were classified in the same cluster.

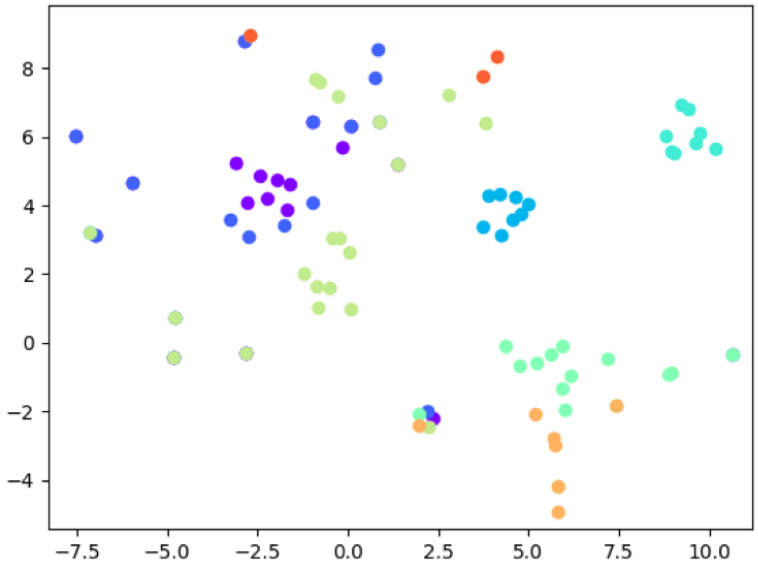


(a) Predicted clusters

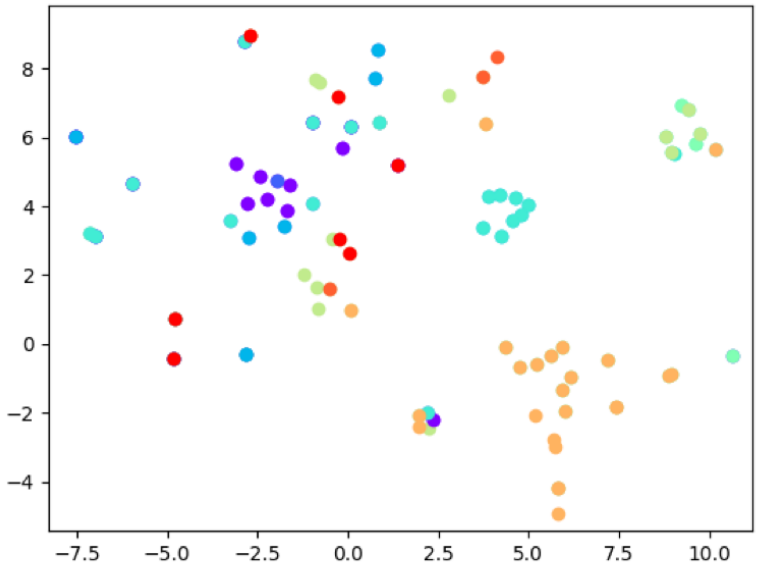


(b) Gold clusters

Figure 4.2: Mention representation for New Plural. The axes represent the mentions' embeddings after applying dimension reduction. All mentions sharing the same color were classified in the same cluster.



(a) Predicted clusters



(b) Gold clusters

Chapter 5

Conclusion

This paper introduces the end-to-end neural plural coreference resolution model and evaluates its performance on the Friends corpus. We first introduce a clear pattern of procedures to adapt coreference resolution models designed for CoNLL'12 to work successfully on the Friends corpus and, more generally, on other corpora with plural speakers and singletons. Then we test different approaches to modify a traditional singular coreference resolution model, such as the end-to-end system by Lee et al. [8], to work for plural coreference resolution and we gradually improve the performance. We explore different labeling techniques and their respective loss functions as well as a variety of clustering algorithms. To the best of our knowledge, we are the first to develop an end-to-end plural coreference resolution model. Our results do not outperform the plural coreference model presented by Zhou and Choi [11]. However, our model did not use any gold mentions at prediction time and the results for our top-performing model $NewPlural + order$ are still within a reasonable margin, especially for the metric Blanc.

Bibliography

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

[2] Henry Y. Chen, Ethan Zhou, and Jinho D. Choi. 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 2017)*, pages 216–225. Association for Computational Linguistics, 2017. doi:

10.18653/v1/K17-1023. URL http://aclweb.org/anthology/K17-1023.

[3] Yu-Hsin Chen and Jinho D. Choi. 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*, pages 90–100. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-3612. URL http://aclweb.org/anthology/W16-3612.

[4] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1245. URL http://aclweb.org/anthology/D16-1245.

[5] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. *CoRR*, abs/1606.01323, 2016. URL http://arxiv.org/abs/1606.01323.

[6] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana, June 2018. Association for

Computational Linguistics. doi: 10.18653/v1/N18-1079. URL http://www.aclweb.org/anthology/N18-1079.

[7] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *CoRR*, abs/1707.07045, 2017. URL http://arxiv.org/abs/1707.07045.

[8] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. *CoRR*, abs/1804.05392, 2018. URL http://arxiv.org/abs/1804.05392.

[9] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 1–40, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2391181.2391183.

[10] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. *CoRR*, abs/1604.03035, 2016. URL http://arxiv.org/abs/1604.03035.

[11] Ethan Zhou and Jinho D. Choi. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on*

Computational Linguistics, pages 24–34. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/C18-1003.