Evaluating Deep Learning Approaches for Character Identification in Multiparty Dialogues
Final Paper: CS585

Sree Harsha Ramesh
Department of Computer Science
University of Massachusetts Amherst
shramesh@cs.umass.edu

Krishna Prasad Sankarayananan
Department of Computer Science
University of Massachusetts Amherst
ksankaranara@cs.umass.edu

Abstract

This paper describes our submission for the SemEval 2018 Task 4 : *"Character Identification on Multiparty Dialogues "*. Our approach for solving this problem has been to model this task as co-reference resolution followed by entity linking for assigning character labels to clusters of named entity mentions. We found that the state-of-the-art co-reference resolution systems performed poorly on the given dataset. However, using an agglomerative convolutional neural network that takes groups of features and learns mention and mention-pair embeddings vastly improved the cluster purity scores for co-reference resolution. The mention embeddings learned were used to create cluster embeddings which were used as inputs to a heuristic entity linker. We were able to achieve a character identification accuracy of 85.28% and an F1-score of 84.41% (on the main 6 characters) on the held-out episodes of the *Friends* dataset.

1 Introduction

Character identification is an entity linking task that identifies each mention as a certain character in multiparty dialogue where mentions are typically nominals referring to a person and entities maybe speakers themselves or even external characters. Identifying such mentions as real characters requires cross-document entity resolution, which makes this task challenging. This task involves coreference resolution which clusters together the mentions corresponding to the same referent followed by an entity linking stage where the clusters of mentions are mapped to their corresponding entities. Historically, coreference models have been trained on the *NewsWire* dataset which is not as rich in terms of the complexity of the coreferences as those in multiparty dialogues. For e.g., the colloquial nature of the dialogue data, such as the one in the shared task, leads to a higher occurrence of anaphors — words that refer to or replace a word used earlier in a sentence, to avoid repetition, such as *do* in *I like it and so do they* — than in the News datasets for which coreference resolution systems are trained. Due to this, coreference resolution systems have been shown not to handle dialogues very well (Niraula et al., 2014), (Rocha, 1999).

However, as has been the norm on various natural language processing tasks, deep-learning models are the state-of-the-art in coreference resolution as well (Clark and Manning, 2016b) and motivates us to extend and evaluate the existing coreference systems including rule-based, statistical and deep-learning based models, for the annotated transcripts of *Friends*, released as part of SemEval 2018 Task-4 [1] described in 5

2 Related Work

The current co-reference resolution systems use a variety of context features along with deep neural architectures to achieve state of the art performance. So, we were motivated to use convolutional neural networks to build embeddings for character identification given that (Wu and Ma, 2017) underline the advantages of CNN in a way that it's apt for integrating individual word semantics in terms of mentions thereby into mention clusters. Although rule based approaches work extremely well for co-reference resolution in specific domains, they have a major pitfall in that the rules need to be heavily hand-crafted.

[1] https://competitions.codalab.org/competitions/17310

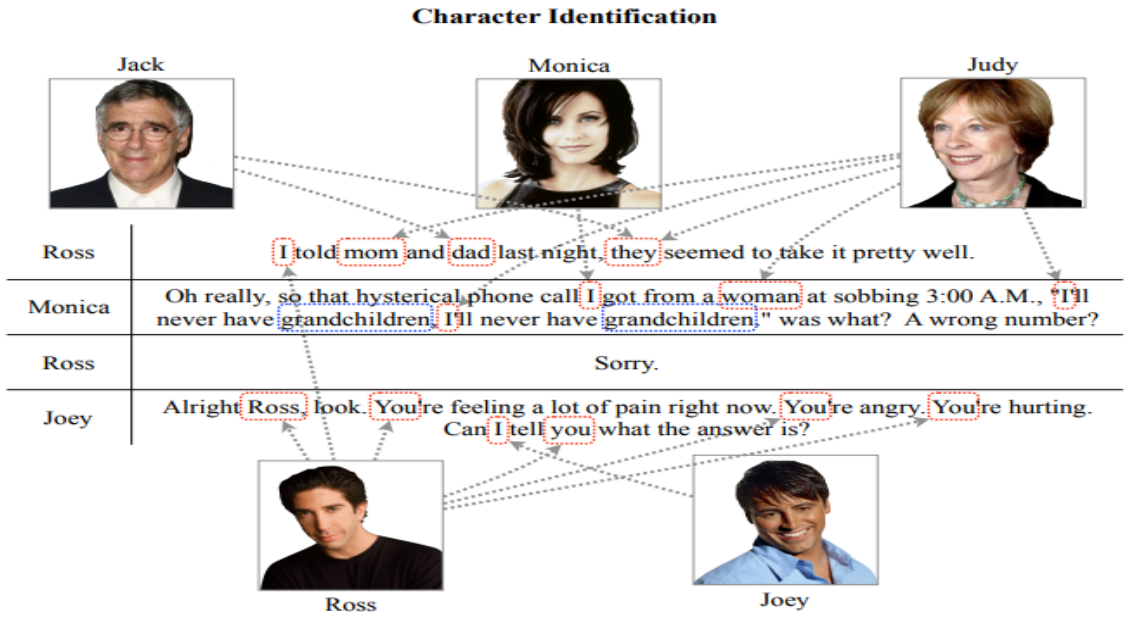

Figure 1: Character identification explained by an example. 3 Characters who are speakers ; Monica, Ross and Joey are being introduced. The aim of character identification is to assign each mention to one of these characters.

Therefore the many tiny relationships between mentions are not easily detected by the model even though it allows for fast prototyping (Lassalle and Denis, 2013). The next set of systems in use are mention ranking models which is based on the mention-pair model but in this case, only the highest coreference score among all the forward antecedents are selected (Stoyanov and Eisner, 2012). Among these set of algorithms, standalone mention identification is usually the go to method for mention identification. Hence our approach uses a classifier to learn the semantic information as well as word dependencies and determines which mention pair has highest coreferent confidence parameter.

(Cheng and Voigt, 2015) propose a new architecture for coreference resolution combining both word-level sequential memory networks and a global level loss function which takes into account pairwise mention-mention links in an entire document. One parameter under consideration was the expectation of pronouns to be operators that can encourage the model to look into it's memory for finding semantically compatible mentions available mentions for the co-reference. This used LSTM for sequential modeling over all the words in the document to generate hidden state vector representations to crate word-level outputs.Complete statistical approach to co-reference has gained favor over the years simply as a log-linear model which take into account features fro state of the art systems.(Durrett and Klein, 2013) point out this by showing that traditional pairwise-style features achieve a good amount of discourse and syntatic variation required fro coreference.

(Chen et al., 2017) propose a deep learning approach to coreference resolution and entity linking for character identification. It introduces a new agglomerative convolutional neural network for returning mention and mention pair embeddings. Entity links are then mapped to each referent separately by cluster embeddings. This method takes into consideration, 20 labels viz. top 9 characters and an unknown label. It emphasizes on the intuition that the coreference resolution accuracy depends upon the size of the clusters. The combined implementation of cluster and mention embeddings improved upon the singular use of mention embeddings in terms of accuracy. Although, agglomerative CNN being an incremental feature approach has its fair share of advantages in terms of word embeddings, the existing approach still lacks the handling of plurals and collective nouns. This scope for overall improvement is aimed to be leveraged by us while looking for extending and evaluating the deep learning approaches.

(Clark and Manning, 2015) introduces an entity centric system using mention pair models as features. Agglomerative clustering is used to build co-reference chains formed by merging pairs of clusters at each step. A key aspect of any incremental co-reference system is its local decisions. Using this to full advantage, costs are assigned to each action which are in turn trained using a cost-sensitive classification. For the ranking model, the current mention is matched to the candidate antecedents simultaneously competing with each other. The resultant prediction model depends upon the previous actions which violate the IID (independent and identically distributed) assumption of statistical learning. Hence, imitation learning is used to classify whether a particular action is the one the expert policy would take at the current state.

(Clark, 2015) proposes a new approach to co-reference resolution using distributed word representations. An incremental co-reference system is defined which acts as a feed forward neural network for mention clusters rather than mention pairs. The usage of mention pairs does not enforce transitivity and therefore relies only on local pairwise information to make co-reference decisions. Mention clusters on the other hand facilitate previous co-reference decisions to inform the latest ones. This is an extension to (Clark and Manning, 2015), wherein features are created between mention clusters using the pairwise probabilities of the mention pair model. This is extended by considering all features from vector representations of mention pairs to produce cluster level features. The actual benefits of deep learning on co-reference are due to the decreased reliance on hand engineered features. (Clark, 2015) demonstrated this by using a simple feature set, yet outperformed the state-of-the-art co-reference systems.

(Wiseman et al., 2015) present a mention ranking model for co-reference resolution. It emphasizes on anaphoricity detection and antecedent ranking with respect to learning feature representations. The training model using backpropagation is preceded by a pre-training segment comprising of two tasks viz. anaphoricity detection and antecedent ranking. The mention ranking model is trained with the slack-rescaled max-margin training objective which facilitates separation between highest scoring true and false antecedents of the current mention. A major challenge of co-reference systems is resolving an anaphoric mention that has no previous head term. This paper intuitively evaluates the possibility of overcoming this challenge by means of non local decision making. It provides a conclusion that pronouns may not be the only coreferent mentions causing these errors and therefore a local model can also be tweaked with respect to a loss function to achieve this.

Having gone through an extensive literature review of the existing approaches for co-reference resolution, we decided to evaluate the deep learning approach for co-reference resolution suggested in (Chen et al., 2017) and compare it against the out-of-the-box coref systems available at Stanford CoreNLP (Manning et al., 2014). Subsequently, we have implemented a heuristic entity linking system to assign character labels to coreferent clusters, thus tackling character identification as an entity linking task.

3 Approach

The overall task of character identification in a multiparty discourse setting, could be divided into two sub-tasks – coreference resolution and entity linking. By integrating the two modules we propose to create a system which richly annotates the dialog data, by mapping mentions to their characters introduced during the discourse.

3.1 Coreference Resolution

Coreference resolution is a task of finding all expressions that refer to the same entity in a text. It plays a crucial role in Natural Language Understanding tasks like document summarization, information extraction and question answering. Figure 2 shows a conversation between three characters of the Friends TV series ; Ross, Joey and Monica.
In the first dialogue, I_1 refers to the speaker Ross. mom_2 and dad_3 are two other mentions referring to Rosss parents. In Joeys dialogue, You_8 refers to who he is speaking to ie Ross.

We would be basing our coreference resolution model on the deep learning approach introduced by (Chen et al., 2017) which involves learning mention and mention pair embeddings using convolutional neural networks. These embeddings

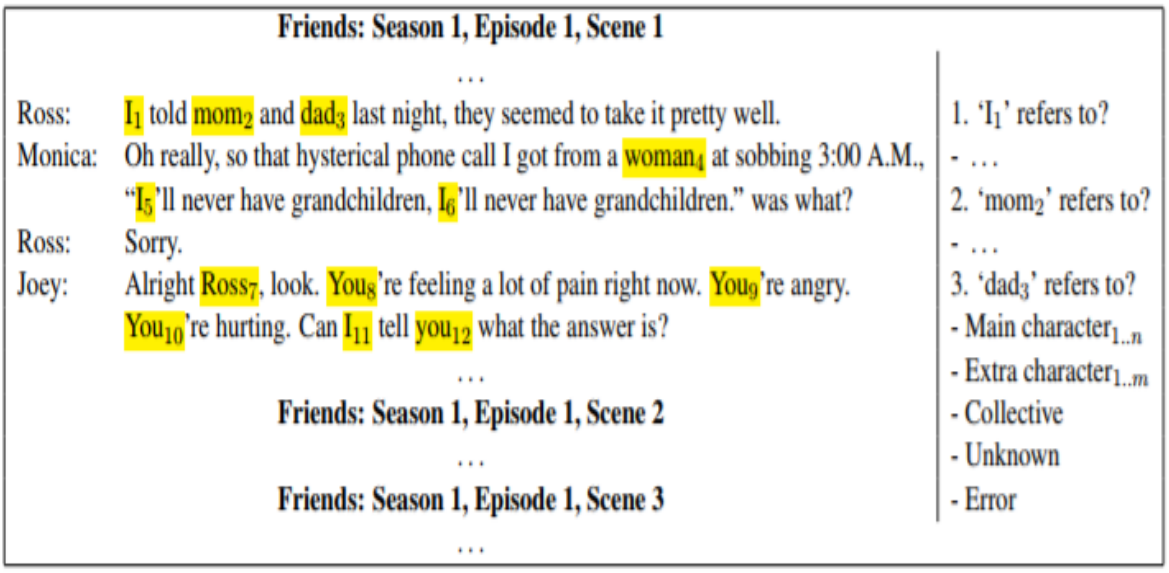

Figure 2: Conversation between characters in the Friends TV Series with mentions being highlighted

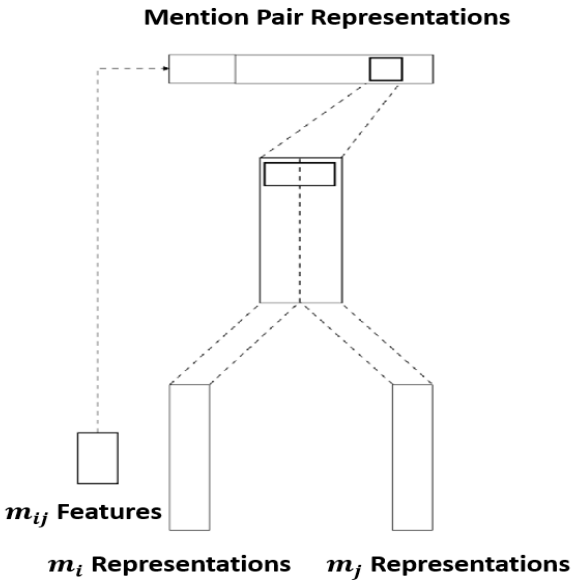

Figure 3: Mention-Pair Representation Model

would be used to get cluster embeddings for the subsequent stage of entity linking.

3.1.1 Mention-Mention Ranking Model

In this paper, we have leveraged the benefits of convolutional neural networks to build a mention-to-mention ranking model. Thereby features which have common properties are segregated into groups and they are trained on separately. This model has resulted in efficient mention and mention pair embeddings.

Feature Extraction

Three main categories of features have been used in this model namely mention embedding, singleton and bi-word features. The word embeddings are trained with Word2vec, Fast text and Glove on Google News, Wikipedia and Amazon reviews. Utterance and sentence vectors are considered to be the weighted average word embeddings of all words in an utterance and a sentence.Plurality and gender data also are based on work from (Bergsma and Lin, 2006) The following table encompasses the entire feature space of the model.

Model

Features extracted as described in the above table are fed into our model at different stages. Consider d_w be the dimension of the mention embedding feature., and f be the number of filters used in each layer.

Pooling and convolution layers are next added into the system architecture. The window sizes of the convolutional layers are given the values 1 x d_w,2 x d_w and 3 x d_w and are trained on each mention embedding feature. The output of each convolution layer is max-pooled alon g the line of the columns. This provides us with a resultant vector of 1 x f by stacking both the convolution and pooling layers into a matrix of dimension 12 x f (Masci et al., 2011). An additional max-pooling and convolution of sizes 1 x f and 12 x 1 respectively are stacked on top of this existing architecture to output a 1 x f vector which will contain the

Feature Group	List of Features
Discrete- ϕ_d	Average plurality information of all words in a mention. Speaker embedding of current and previous utterance. Average gender information of all words in a mention. Average word animacy of all the words in m
Pair-wise Features - ϕ_p	Speaker information of the mention pair Longest Common Subsequence of words in mentions Common words between mentions. Distance and position metrics between mentions. Sentence and mention distance between m_i and m_j. Speaker match between m_i and m_j.
Mention Embedding - ϕ_m	Utterance vectors of current,previous and successor utterances Sentence vectors of current,previous and successor mentions before utterances. Average word embeddings of all words in a mention. Words embeddings of n preceeding and succeeding words in a mention.

Table 1: Feature Template

categorized mention embedding features. Mention representations are the culmination of these mention embedding features by concatenating and flattening the vectors of the convoluted embeddings. For every mention, this encapsulates the local context of the given mention.

Consider $r_m(mi)$ to be the mention representation of mention m_i of dimension d_{rm}. The mention pair representation model shown in **??** contains mention representations $r_m(m_i)$ and $r_m(m_j)$ which are stacked into a 2 x d_{rm} matrix. A mention pair vector of dimension f is created by applying a single convolution layer and max-pooling layers of size $1xd_{rm}$ and $2x1$. Mentions m_i and m_j are used to extract pairwise mention features which are in turn concatenated with this vector. As a result, $r_{mm}(m_i, m_j)$ are the mention pair representations of m_i and m_j as shown in the below figure.

3.2 Mention-Mention Ranking Model

We use a scoring function s_{mm} for determining the likelihood of a link between the two mentions given its mention pair representation $r_{mm}(m_i, m_j)$ between m_i and m_j

$$s_{mm}(m_i, m_j) = \sigma(W_{mm} r_{mm}(m_i, m_j) + b_{mm}) \quad (1)$$

Here w_{mm} and b_{mm} are the weights and bias of the scoring function. Th scoring function is essentially a regression model used to train our model with a mean squared error loss function. Let $A(m_i)$ be the list of antecedents of each mention m_i and $C(m_j)$ be the cluster containing the mention. For each mention , the goal is to find the training instances up to the closes antecedent with a linking score of 1. This condition for the gold linking score $p(a, m)$ is given as follows.

$$p(a, m) = \begin{cases} 1 & \text{if } m \in C(a) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Through back-propagation of the loss function, the model learns mentions and mention pair representations which in turn optimizes the task of mention ranking.

3.2.1 Agglomerative CNN

The model is called agglomerative due to the way in which it generates mention clusters iteratively, agglomerating mentions into clusters along the way by using multiple feature groups in convolutional layers. This results in creating better generate mention and mention pair embeddings. The rest of the section describes the architecture of the ACNN model,

Considering two feature maps $\phi_e^k(m)$ and $\phi_d^k(m)$ where m is the mention, $\phi_e^k(m)$ extracts feature embeddings based on words and $\phi_e^d(m)$ extracts standalone stand alone features. Each embedding group k is applied upon a convolutional layer $CONV_1^k$ with n-gram filter of size d. The resultant vector is max-pooled to generate a feature vector $\epsilon \mathbb{R}^{1xd}$. The second convolutional layer $CONV_2^k$ is applied on a 3 dimensional feature matrix $\epsilon \mathbb{R}^{nxdxk}$ from the previous convolutional

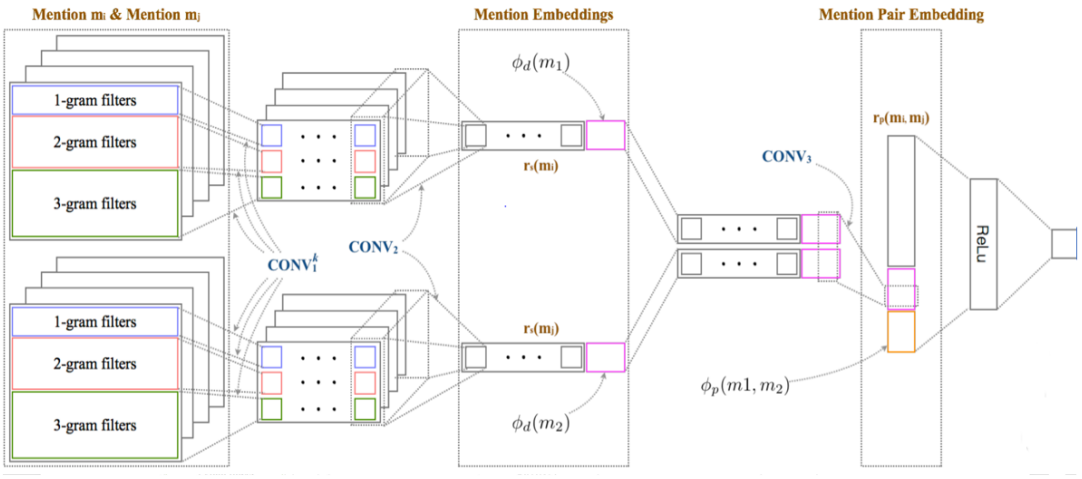

Figure 4: Mention pair embedding using Agglomerative CNN

layer. This result is concatenated and max-pooled with the extracted standalone features $\phi_e^d(m)$ to from mention embedding $r_s(m_i)$ defined as follows:

$$\mathbf{r}_s(m) = CONV_2(\begin{bmatrix} CONV_1^1(\phi_e^1(m)) \\ \vdots \\ CONV_k^1(\phi_e^k(m)) \end{bmatrix})||\phi_d(m) \tag{3}$$

This leads us to the next step to convert mention embeddings to mention pair embeddings. The other feature type in the feature space are the bi-word features $\phi_o p(m_i, m_j)$. The third convolutional layer $CONV_3^k$ is applied on mention embeddings $r_s(m_i)$ and $r_s(m_j)$. This result is concatenated and max-pooled with the bi-word features to form $r_p(m_i, m_j)$ defined as follows:

$$\mathbf{r}_p(m_i, m_j) = CONV_3(\begin{bmatrix} \mathbf{r}_s(m_i) \\ \mathbf{r}_s(m_j) \end{bmatrix})||\phi_p(m_i, m_j) \tag{4}$$

The learned mention pair embeddings are passed through a hidden layer with ReLu and sigmoid function $\sigma(m_i, m_j)$ to determine the coreferent relation between m_i and m_j defined as follows:

$$\begin{aligned} h(x) &= ReLU(w_h x + b_h) \\ \sigma(m_i, m_j) &= sigmoid(w_s h(r_p(m_i, m_j)) + b_s) \end{aligned} \tag{5}$$

For each mention, $\sigma(m_i, m_j)$ performs binary classifications between m_i and m_j. It considers a halfway threshold and the model considers no coreferent relation between m_i and m_j if it's below the threshold and therefore creates a new cluster with it.

The rules for mention clusters are defined as follows:

- If $\forall_{1 \le j \le i}.max(\sigma(m_i, m_j)) < 0.5$, then create a new cluster C_{m_i}.
- If $\exists_{1 \le j \le i}.max(\sigma(m_i, m_j)) \ge 0.5$, then $1.C_{m_k} \leftarrow C_{m_k} \cup m_i$, $2.m_k = arg_j max(\sigma(m_i, m_j))$.

3.3 Entity Linking

In the previous stage, coreference resolution groups mentions into clusters, but it does not assign character labels to the clusters, which is required for character identification. Thus, an entity linking model is required that takes the mention embeddings and the mention-pair embeddings generated by the CNN and classifies each mention to one of the character labels. This would involve creating a heuristic entity linking system for cluster remapping.

3.3.1 Rule-based entity linker

Theoretically, the larger a document is the large its coreferent chains are, and in the context of our task, the clusters from running coreference resolution on entire TV show should each represent an

Train/Dev/Test	Episodes	Scenes	Mentions
Train - F1 + F2	37	284	10364
Train - B1	14	76	3565
Dev - F1 + F2	5	38	1357
Dev - B1	1	7	390
Test - F1 + F2	5	52	1827
Test - B1	2	12	551

Table 2: F1 - Friends Season 1, F2 - Friends Season 2, B1 - Big Bang Theory Season 1

entity. However, it is nearly impossible to have such a perfect system. The resultant chains are often segmented and noisy, therefore they are less obvious in the specific characters they are referring to. Since the predicted coreferent chains do not directly point to specific entities, a mapping mechanism is needed for linking those chains to certain characters.

We have implemented a rule-based entity linker that evaluates and pools the mention within found coreferent chains. The resultant chains from systems are mapped to either a character, collective, or unknown entity. Each coreference chain is reassigned through voting on the majority entity assignment of mentions. The referent of each mention is determined by the below high-precision rules:

- If the mention is a proper noun or a named entity that refers to a known character, it is referent to the character.
- If the mention is a first-person pronoun or possessive pronoun, it is referent to the speaker character of the utterance containing the mention.
- If the mention is a collective pronoun or collective possessive pronoun, it is referent to the collective group.

If none of these rules apply to any of the mentions in a coreference chain, the chain is mapped to the unknown group.

4 Dataset

In this section, we would be describing the dataset released as part of SemEval-2018, which would be used for training and dev-testing the character identification system. The validation dataset would be released in January 2018, so we have held back some data during the training, to report the accuracy on. The train/test/dev splits can be seen in Table 2.

The 2018 dataset comprises first two seasons of the TV show - Friends, annotated for this task. We have also used one season of the TV show Big Bang Theory [2] released by the shared-task organizers in 2016. Each season consists of episodes, each episode comprises scenes, and each scene is segmented into sentences. The corpus statistics for the entire dataset available can be seen in Table 5.

4.1 Data Format

All datasets follow the format of the data released for the CoNLL 2012 shared task (Pradhan et al., 2012). We have listed all the columns that are typically in a co-reference resolution shared task, including those that were not provided for this task such as the *frameset id* and the *word sense*. The following are the columns for every token in an utterance.

1. *Document ID*: name of the show-season ID-episode ID (e.g., friends-s01e01, or bigbang011001).
2. *Scene ID*: the ID of the scene within the episode.
3. *Token ID*: the ID of the token within the sentence.
4. *Word form*: the tokenized word.
5. *Part-of-speech tag*: the part-of-speech tag of the word (auto generated).
6. *Constituency tag*: the Penn Treebank style constituency tag (auto generated).
7. *Lemma*: the lemma of the word (auto generated).
8. *Frameset ID*: not provided.
9. *Word sense*: not provided.
10. *Speaker*: the speaker of this sentence.
11. *Named entity tag*: the named entity tag of the word (auto generated).
12. *Entity ID*: the entity ID of the mention, that is consistent across all documents.

[2] https://github.com/emorynlp/character-mining/blob/master/md/corpus.md

Stanford Coref Model	Document:episode_delim				Document:scene_delim			
	MUC	$CEAF_e$	B^3	μ	MUC	$CEAF_e$	B^3	μ
Clarke and Manning (2016) Mention -Mention	13.81	13.81	13.81	13.81	34	34	34	34
Clarke and Manning (2016) Coreference	8.87	2.2	2.88	4.65	26.65	12.48	16.02	18.38
Wiseman et al. Mention-Mention (2016)	18.39	18.39	18.39	18.39	47.6	47.6	47.6	47.6
Wiseman et al. Coreference (2016)	13.08	3.51	4.01	6.86	39.89	18.29	23.99	27.39

Table 3: Cluster purity scores for coreference resolution using Stanford DeepCoref and Sieve based models.

Training Set	Pretrained Embeddings	Document-type	Precision	Recall	F1
f1+f2	Glove	Scene-delim	0.8458	0.5408	0.6598
f1+f2	fastText	Scene-delim	0.7382	0.6575	0.6955
f1+f2	w2v	Scene-delim	0.7821	0.6518	**0.7110**
f1+f2	w2v	Episode-delim	0.8238	0.4795	0.6062
f1+f2+b1	w2v	Scene-delim	0.7938	0.7206	0.7554
f1+f2+b1	w2v	Episode-delim	0.7973	0.3734	**0.5086**

Table 4: Cluster purity scores for coreference resolution using our model

TV Shows	Season	Episode	Scene	Speaker	Utterance	Statement	Word	Mention
Friends	1	24	194	96	4,025	7,423	56,691	6,837
	2	23	180	88	3,696	6,126	54,713	6,711
Big Bang Theory	1	17	95	31	2,392	3,302	33,835	4,506

Table 5: Corpus Statistics

Redundant Entity Id	Character Name	Actual entity id (modified)	Incorrectly Annotated Phrases
724	Leslie	723	Leslie Winkle (Full name considered a new entity)
731	Leslie	723	See *you* next week (Incorrect speaker resolution)
751	Leslie	723	Leslie Winkle (Full name considered a new entity)
752	Leslie	723	Leslie . . . (Boundary error - ellipsis)
732	Leonard	713	Leonard's (Boundary error - possession)
745	Leonard	713	Leonard . . . (Boundary error - ellipsis)
746	Leonard	713	Leonard Hofstadter (Full name considered a new entity)
739	Sheldon	714	Sheldon Cooper (Full name considered a new entity)
748	Sheldon	714	Sheldon . . . (Boundary error - ellipsis)
728	Howard	719	Howard Wolowitz (Full name considered a new entity)
734	Penny	717	Penny's (Boundary error - possession)
738	Howard's Mother	737	Excuse *me* (Incorrect speaker resolution)
740	Lalita	741	Lalita Gupta (Full name considered a new entity)
750	Dennis	749	Dennis Kim (Full name considered a new entity)

Table 6: A sample of corpus annotation errors in *Big Bang Theory* along with the error types, and corrections.

4.2 Corpus Annotation Errors

During the course of working on this project, we found quite a few annotation errors in the dataset, especially on the *Big Bang Theory* dataset. In Table 6 , we present a qualitative analysis of the kind of errors made by the Mechanical Turkers while assigning the character labels to mentions in the dialogue corpus. Most common error is the incorrect resolution of the full names of a given character. Also included in the table are boundary errors where stray punctuation symbols such as ellipsis, apostrophe are added to the named entity phrase and considered a new entity. These problems are probably very commonplace when annotation is crowd-sourced.

5 Experiments and Evaluation

In this section, we present the results of the different experiments that were conducted to test the deep learning model of co-reference resolution against the baseline. Since, we found that the coref-resolution results of the Stanford models 3 were very poor compared to the agglomerative CNN, we chose to report the results of entity linking on the latter.

5.1 Coreference Evaluation Metrics

The 4 systems trained as part of the preliminary experiments, were all evaluated with the official CoNLL scorer[3] on the three metrics for measuring coreference resolution: MUC, B^3, $CEAF_e$. The results can be seen in Table 1.

5.1.1 B^3

According to (Bagga and Baldwin, 1998), instead of evaluating solely on the coreference chains, the B^3 metric calculates precision and recall values on a mention level basis. The average of all these mention scores is the performance of the system. Hence, for a set M containing mentions m_i, consider coreference chains S_{m_i} and G_{m_i} to be pertaining to system and gold responses respectively. Precision - P and Recall - R are calculated as follows:

$$P(m_i) = \frac{|S_{m_i} \cap G_{m_i}|}{|S_{m_i}|} \tag{6}$$

[3]https://github.com/conll/reference-coreference-scorers

$$R(m_i) = \frac{|S_{m_i} \cap G_{m_i}|}{|g_{m_i}|} \tag{7}$$

5.1.2 MUC

This metric proposed by (Vilain et al., 1995) takes into account the number of pairwise links with respect to the gold keys. Precision is calculated by dividing the number of links the system shares with the gold keys and the minimum number of links needed to describe coreference chains of gold and the system.

5.1.3 CEAF

This metric proposed by (Luo, 2005) is an improved version of B^3 which had a pitfall that entities could be used more than once during evaluation. As a result, chains with same entity and multiple entity mention chains are not taken into account for. Hence, to overcome this CEAF outputs the best one-to-one mapping between gold and system predicted entities ie it gives the count of common mentions that pertain to both system and gold labels. This entity based similarity is referred to as $\phi(S_i, S_j)$ or gold G_i and System S_i. The best similarity measure is denoted as $\phi(g^*)$. Thereby Precision P and Recall R are calculated as follows:

$$P = \frac{\phi(g^*)}{\sigma_i \phi(S_i, S_j)} \tag{8}$$

$$R \frac{\phi(g^*)}{\sigma_i \phi(G_i, G_j)} \tag{9}$$

5.2 Baseline

Since the precursor to character identification is identification of entity mention clusters, we experimented with two of the open-source state-of-the-art coreference systems which identify coreferant words or phrases in the discourse, that could later be linked to characters based on some characteristics of the mentions in a group.

5.2.1 Stanford Deterministic Coreference Resolution System

This is based on the multi-pass sieve system proposed by (Lee et al., 2013). It is composed of multiple sieves of linguistic rules that are in the orders of high-to-low precision and low-to-high recall. Information regarding mentions, such as plurality, gender, and parse tree, is extracted during mention detection and used as global features.

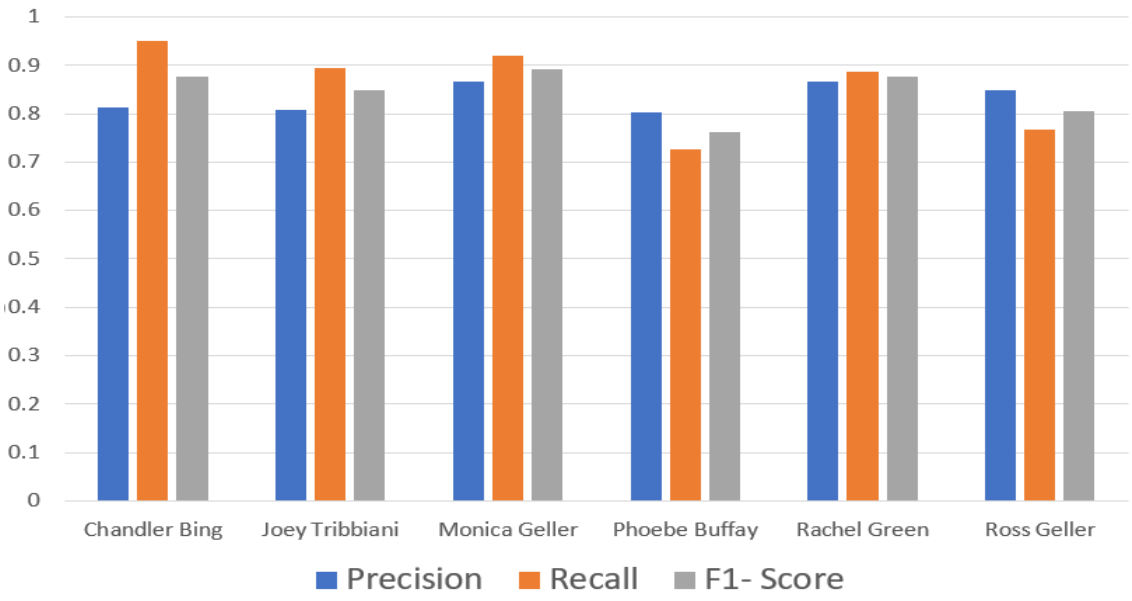

Figure 5: Character identification axccuracy

Pairwise links between mentions are formed based on defined linguistic rules at each sieve in order to construct coreference chains and mention clusters. We created episode-delimited and scene-delimited documents and ran the rule-based deterministic coreference resolution system[4].

5.2.2 Stanford Neural Coreference Resolution System

This system is based on the neural coreference system proposed in (Clark and Manning, 2016a) and (Clark and Manning, 2016b). We used the pre-trained models which were trained on dataset released as part of the ConLL-2012 shared task [5]. Thus, the dense vector representations learned for mention pairs in the CoNLL dataset were used to model the entity level information on our dataset. It uses minimal number of hand-engineered features compared to the rule-based system described in Section 4.3.1.

5.3 Our Model

We trained the ACNN model across both scene-delimited and episode-delimited transcripts using the above described architecture. We played around with different pre-trained embeddings such as Word2Vec (Mikolov et al., 2013) embeddings trained on Google News Dataset, FastText (Bojanowski et al., 2016) embeddings trained on Wikipedia, and Glove (Pennington et al., 2014) vectors trained on Wikipedia, and found that the best accuracy was found when we used the Word2Vec embeddings.

Table 4, lists the F1-scores for different experiments conducted using measured using the B^3 cluster purity metric for the coref-resolution system trained using the Agglomerative CNN and the Table 3 shows the out-of-the-box cluster purity scores from the Stanford Coreference models, which were used as the baseline. By comparing the two we can see that our model almost always beats the baseline.

As required by the shared task submission, we are reporting the following metrics that were obtained on the best performing model that was trained on scene-delimited documents and word2vec embeddings:

1. The label accuracy considering only 7 entities, that are the 6 main characters (Chandler, Joey, Monica, Phoebe, Rachel, and Ross) and all the others as one entity is **85.28%**

2. The macro average between the F1 scores of the 7 entities. **84.41%**

3. The F1 scores for 7 entities are listed in the Figure 5

6 Conclusion

In this paper, we have built a character identification system for multi-party dialogues. We have

[4]https://nlp.stanford.edu/software/dcoref.html
[5]http://conll.cemantix.org/2012/data.html

developed a neural approach to coreference resolution using agglomerative CNN which aggregates the feature groups into mention, mention pair representations, cluster and mention-cluster embeddings. A heuristic entity linker has been proposed for cluster mapping. We have used three pretrained embeddings ; Word2vec on Google News, fastText and glove on English Wikipedia and have calculated the cluster purity scores for scene and episode delimited documents on the Friends and Big Bang theory transcripts. We found out that our model best worked on scene delimited documents with an F1 score of 0.7554. Also, we got a character identification accuracy of 85.28% on the 6 main characters of Friends. In future, we would like to extend our approach towards improving the embeddings to take into account collective and plural mentions. Also, we would experimenting with the embeddings trained on Amazon product reviews (He and McAuley, 2016). Another major addition would be to build a feed forward neural network for entity linking.

7 Acknowledgment

We would like to acknowledge the organizers of the SemEval-2018 shared task for curating this very interesting task and for releasing the benchmarking implementation of the ideas in the papers (Chen and Choi, 2016) (Chen et al., 2017). We look forward to submitting a run at this at this year's SemEval, once the test data is released!

References

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

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

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

Henry Y 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 2017)*. pages 216–225.

Yu-Hsin Chen and Jinho D Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *SIGDIAL Conference*. pages 90–100.

Xiao Cheng and Rob Voigt. 2015. A deep architecture for coreference resolution. In *Proc. of the 2001 Workshop on Computational Natural Language*. pages 1–8.

Kevin Clark. 2015. Neural coreference resolution.

Kevin Clark and Christopher D Manning. 2015. Entity-centric coreference resolution with model stacking. In *ACL (1)*. pages 1405–1415.

Kevin Clark and Christopher D Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667* .

Kevin Clark and Christopher D Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323* .

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *EMNLP*. pages 1971–1982.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pages 507–517.

Emmanuel Lassalle and Pascal Denis. 2013. Improving pairwise coreference models through feature space hierarchy learning. In *ACL 2013-Annual meeting of the Association for Computational Linguistics*.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics* 39(4):885–916.

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

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning–ICANN 2011* pages 52–59.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Nobal B Niraula, Vasile Rus, Rajendra Banjade, Dan Stefanescu, William Baggett, and Brent Morgan. 2014. The dare corpus: A resource for anaphora resolution in dialogue based intelligent tutoring systems. In *LREC*. pages 3199–3203.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*. Association for Computational Linguistics, pages 1–40.

Marco Rocha. 1999. Coreference resolution in dialogues in english and portuguese. In *Proceedings of the Workshop on Coreference and its Applications*. Association for Computational Linguistics, pages 53–60.

Veselin Stoyanov and Jason Eisner. 2012. Easy-first coreference resolution. In *COLING*. pages 2519–2534.

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

Sam Joshua Wiseman, Alexander Matthew Rush, Stuart Merrill Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. Association for Computational Linguistics.

Jheng-Long Wu and Wei-Yun Ma. 2017. A deep learning framework for coreference resolution based on convolutional neural network. In *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*. IEEE, pages 61–64.