Neural Anaphora Resolution in Dialogue

Hideo Kobayashi* and **Shengjie Li*** and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{hideo,sxl180006,vince}@hlt.utdallas.edu

Abstract

We describe the systems that we developed for the three tracks of the CODI-CRAC 2021 shared task, namely entity coreference resolution, bridging resolution, and discourse deixis resolution. Our team ranked second for entity coreference, first for bridging resolution, and first for discourse deixis resolution.

1 Introduction

The CODI-CRAC 2021 shared task (Khosla et al., 2021), which focuses on anaphora resolution in dialogue, provides three tracks, namely entity coreference resolution, bridging resolution, and discourse deixis/abstract anaphora resolution. While the CRAC 2018 shared task (Poesio et al., 2018) provides the same three tracks, the two shared tasks differ by the genre they focus on: CRAC 2018 focuses primarily on text, whereas CODI-CRAC 2021 focuses exclusively on spoken dialogue.

Not only has entity coreference resolution been an active area of research in the NLP community in the past few decades, but recent years have seen considerable progress on entity coreference because of the development of span-based neural models (Lee et al., 2017, 2018). Compared to entity coreference, bridging resolution and discourse deixis resolution are much less studied, and hence they are arguably the more interesting tracks of this shared task. In particular, a relevant question is: can the successes of span-based models be extended from entity coreference to bridging resolution and discourse deixis resolution?

We participated in all three tracks of the shared task. For bridging and discourse deixis resolution, we submitted results based on both predicted mentions and gold mentions. Given the recent successes of span-based neural entity coreference models, which can learn task-specific representations of text spans, we use them as our starting point for all three tracks. Specifically:

- for entity coreference, we employ a pipeline architecture where we perform mention detection prior to coreference resolution. Our mention detection model is adapted from Xu and Choi's (2020) implementation of Lee et al.'s span-based model. For coreference resolution, we extend Xu and Choi's coreference model by (1) adding a sentence distance feature; (2) modifying the objective function used by the model so that it can learn to output singleton clusters; and (3) introducing non-coreference constraints for the dialogue domain.
- for discourse deixis resolution, we extend Xu and Choi's coreference model by (1) modifying the objective function so that it performs joint mention detection and discourse deixis resolution, and (2) classifying a span as a candidate anaphor or a candidate antecedent and only allowing candidate anaphors to be resolved to candidate antecedents.
- for bridging, we adopt a multi-pass sieve approach, where we use Yu and Poesio's (2020) multi-task learning (MTL) model, which jointly identifies bridging and coreference links, as one of the sieves and design a set of sieves that targets a particular kind of bridging links, namely same-head bridging links.

A brief overview of the approaches we adopted for the three tracks can be found in Table 1.

The rest of the paper is structured as follows. The next three sections describe our work for the three tracks, namely entity coreference (Section 2), discourse deixis (Section 3), and bridging (Section 4). Within each section, we describe our approach, our official results, and some quantitative analysis. We present our conclusions in Section 5.

*Equal contribution

Entity Coreference Resolution	
Baseline	Xu and Choi's (2020) implementation of Lee et al.'s (2018) span-based model
Learning framework	A pipeline architecture consisting of a mention detection component and an entity coreference component. The coreference component extends the baseline by (1) adding a sentence distance feature; (2) modifying the objective so that it can output singleton clusters; and (3) enforcing dialogue-specific non-coreference constraints.
Markable identification	A mention detection model (adapted from Xu and Choi's coreference model) is trained to identify the entity mentions.
Training data	90% of the official training and dev sets
Development data	The remaining 10% of the official training and dev sets
Discourse Deixis Resolution	
Baseline	Xu and Choi's (2020) implementation of Lee et al.'s (2018) span-based model
Learning framework	Joint mention detection and coreference resolution enabled by modifying the objective function in Xu and Choi's model. For mention detection, each span is classified as a candidate anaphor, a candidate antecedent, or a non-mention. For deixis resolution, only candidate anaphors will be resolved, and they can only be resolved to candidate antecedents. The model developed for the Predicted setting differs from those developed for the Gold setting in terms of the heuristics used to determine which spans are candidate anaphors.
Markable identification	Obtained as part of joint mention detection and deixis resolution
Training data	Two setups: (1) use all official training and dev sets, leaving out the official dev set of the target domain; and (2) use 90% of the official training and dev sets.
Development data	Two setups: (1) use only the dev set for the target domain; and (2) use the remaining 10% of the official training and dev sets.
Bridging Resolution	
Baseline	Yu and Poesio's (2020) multi-task learning (MTL) model
Learning framework	A multi-pass sieve approach in which we use the MTL model as one of the sieves and design a set of learning-based sieves (trained using SVMs) that targets same-head bridging links. In the Gold setting, an anaphor detection model (adapted from the MTL model) is additionally trained to first identify the bridging anaphors from the gold mentions, and the resulting anaphors are then resolved by the sieves.
Markable identification	The MTL-based sieve learns markable boundaries whereas the SVM-based sieves use extracted NPs. In the gold setting, an anaphor detection model is also trained to identify bridging anaphors.
Training data	The MTL-based sieve is pretrained using ARRAU RST (train, dev, and test), Gnome, and Pear, and then fine-tuned using Trains 91, Trains 93, LIGHT (dev), AMI (dev), Persuasion (dev), and Switchboard (dev). The SVM-based sieves are trained using LIGHT (dev), AMI (dev), Persuasion (dev), and Switchboard (dev).
Development data	Trains 91, Trains 93, LIGHT (dev), AMI (dev), Persuasion (dev), and Switchboard (dev)

Table 1: Overview of the approaches we adopted for the three tracks.

2 Entity Coreference Resolution

To build our entity coreference system, we use as our baseline *coref-hoi*, which is Xu and Choi's (2020) coreference model. Below we first give an overview of coref-hoi and then describe our extensions to it.

2.1 Baseline: An Overview

Xu and Choi (2020) reimplement the end-to-end coreference model introduced by Lee et al. (2018). For each mention span x, the model learns a distribution $P(y)$ over possible antecedents $y \in \mathcal{Y}(x)$:

$$P(y) = \frac{e^{s(x,y)}}{\sum_{y' \in \mathcal{Y}(x)} e^{s(x,y')}}$$

where $s(x, y)$ is a pairwise score that incorporates three factors: (1) $s_m(x)$, a score that indicates how likely span x is a mention; (2) $s_m(y)$, a score that indicates how likely span y is a mention; and (3) $s_c(x, y)$, a score that indicates how likely spans x and y refer to the same entity:

$$\begin{aligned} s(x,y) &= s_m(x) + s_m(y) + s_c(x,y) \\ s_m(x) &= \texttt{FFNN}_m(g_x) \\ s_c(x,y) &= \texttt{FFNN}_c(g_x, g_y, \phi(x,y)) \end{aligned}$$

where g_x and g_y denote the span embeddings of x and y, $\texttt{FFNN}(\cdot)$ denotes a feedforward neural network, and $\phi(x, y)$ encodes the speaker information from the metadata as well as the segment distance between the two spans.[1]

Xu and Choi describe several higher-order inference (HOI) approaches that can be added to the basic end-to-end coreference model. We do not employ any HOI approaches because (1) Xu and Choi found that when using SpanBERT as the encoder, the impact of HOI is negative to marginal;

[1] Each document is split into independent segments with a maximum size of 512 tokens.

(2) in preliminary experiments, we found that better results could be achieved without HOI.

2.2 Approaches

Next, we describe two approaches to entity coreference resolution.

2.2.1 End-to-End Approach

We extend the aforementioned coref-hoi model with the following modifications.

Sentence distance We hypothesize that recency plays a role in resolution, so we add the sentence distance between two spans into $\phi(x, y)$ as another feature.

Type prediction Since the official scorer penalizes a mention e in the system output if e is not a referring entity mention, we follow our previous work (Lu and Ng, 2020) and extend the model so that it can predict the type of each span, where the type can be NULL (for non-entity spans), REFERRING (for referring entity mentions), or NON-REFERRING (for non-referring entity mentions), and subsequently remove from the output any spans that are predicted to be NULL or NON-REFERRING.

Type prediction proceeds as follows. For each span x, we pass its representation g_x to a FFNN, which outputs a vector ot_x of dimension 3. Each element $ot_x(t)$ of ot_x denotes the likelihood that span x belongs to type t. The span type t_x is then determined by the type with the highest score.

$$ot_x = \text{FFNN}_t(g_x)$$
$$t_x = \arg\max_t ot_x(t)$$

We compute the cross-entropy loss using ot_x. This type loss is then multiplied by a type loss coefficient and added to the loss function of coref-hoi.

Span constraint While span-based models typically impose length constraints on spans owing to computational tractability, we impose an additional constraint on spans based on our observation of the training and development data: a span cannot cover more than one speaker's utterances.

Resolution constraints We propose a consistency constraint on resolution that will be used in both training and inference. This constraint prevents two spans x and y that both start with a pronoun from being posited as coreferent if they are *conflicting*. More specifically, we check whether each of these spans belongs to one of the eight groups defined in Table 2, and if yes, then they cannot be coreferent if any of the following conditions is satisfied:

Group	Definition
1	span is or starts with: I, me, my, or mine
2	span is or starts with: you, your, or yours
3	span is or starts with: he, him, or his
4	span is or starts with: she or her
5	span is: their
6	span is: it or its
7	span is: here
8	span is: there

Table 2: The eight groups of spans on which the consistency constraints for entity coreference resolution are defined.

- Both spans (1) belong to the first four groups but are not in the same group and (2) have the same speaker.
- Both spans (1) belong to the first two groups and are in the same group and (2) have different speakers.

In addition, we impose two constraints on the resolution of *here* and *there*. Specifically, *here* cannot be coreferent with *my*, *your*, *his*, and *her* as well as a span that belongs to group 5, group 6 or group 8; and *there* cannot be coreferent with *my*, *your*, *his*, and *her*, as well as a span that belongs to group 5, group 6, or group 7. These conditions are derived based on our inspection of the training and development sets.

2.2.2 Pipeline Approach

In the end-to-end approach, type prediction is learned jointly with entity coreference resolution. In this subsection, we experiment with a *pipeline* approach where type prediction is performed prior to coreference resolution.

Step 1: Type prediction. The goal of this step is to identify the entity mentions (including both referring and non-referring mentions) from all the spans in the input text. Specifically, the model classifies each span as one of two types, NULL and ENTITY, where ENTITY covers both referring and non-referring mentions and NULL covers the remaining spans (i.e., the spans that do not correspond to entities). To perform this step, we train coref-hoi (with the extensions described in Section 2.2.1), but tune the type loss coefficient so that the model focuses on type prediction rather than coreference resolution.

Step 2: Coreference resolution. The goal of this step is to perform coreference resolution on all and only those spans that are classified as ENTITY in the first step. To perform this step, we train coref-hoi (with the extensions described in Section 2.2.1) on only the gold mentions in the input documents, tuning the type loss coefficient so that the model focuses on coreference resolution rather than type prediction.

2.3 Evaluation

We evaluate the pipeline approach and the end-to-end approach. In addition, to gain insights into the contribution of the constraints on performance, we evaluate a variant of the end-to-end approach without using the span and consistency constraints.

2.3.1 Training and Development Sets

We experiment with two different methods for partitioning the available annotated data into a training set and a development set.

`T1`**:** In the first method, we use all official training datasets and all official development sets other than the one to be evaluated on as our training data. The remaining official development data is then used for development. For example, when we train the model for evaluation on LIGHT_test, we use all official training data plus AMI_dev, Persuasion_dev, and Switchboard_dev as the training set, and use LIGHT_dev as the development set.

`T2`**:** In the second method, we merge all official training sets and all official development sets into one dataset. We then use 90% of this dataset for training and the remaining 10% for development.

2.3.2 Implementation Details

In all approaches we use SpanBERT$_{\text{Large}}$ as the encoder. Documents are split into independent segments with a maximum of 512 word pieces, and two segments from each document are used in training. We use different learning rates for BERT-parameters and task-parameters (1×10^{-5} and 3×10^{-4} respectively). Models are trained for 24 epochs with dropout rate 0.3. The type loss coefficient is found using grid search. See Appendix A for the optimal hyperparameters chosen for each approach.

For the end-to-end approach (and its "no constraint" variant), we use method `T1` to create the training and development sets. The model considers all spans that have a width of less than 70 *and* satisfy the span constraint mentioned in Section 2.2.1 (with the exception of the AMI dataset, where we set the maximum span width to 25 during inference because of memory limitations). To maintain computational tractability, the model selects for each document $0.4 \times \#\ of\ words\ in\ the\ document$ spans as top spans for further processing. Specifically, for each top span, the model selects 50 candidate antecedents for resolution purposes (with the exception of the AMI dataset, where we use only 12 candidate antecedents during inference because of memory limitations). The only parameter we tune is the type loss coefficient. Specifically, we search for the coefficient out of {0, 0.2, 0.4, 0.5, 0.6, 0.8, 1, 2}, and apply the model with the highest CoNLL score on the development set to the test sets.

For the pipeline approach, we use method `T2` to create the training and development sets. We set the parameters of the type prediction model to be the same as those used in the end-to-end approach. Specifically, it considers all spans with a width of less than 70 (with the exception of the AMI dataset, where we set the maximum span width to 25 during inference because of memory limitations), and selects $0.4 \times \#\ of\ words\ in\ the\ document$ spans as top spans for further processing. For each top span, the model selects 50 candidate antecedents for resolution purposes (with the exception of the AMI dataset, where we use only 12 candidate antecedents because of memory limitations). The only parameter we tune is the type loss coefficient. Specifically, we search for the coefficient out of {1, 2, 5, 10, 100, 200, 500}, and use the model with the highest entity mention recall on the development set to predict entity mentions in the test sets. Like in the type prediction model, the only parameter we tune in the coreference model is the type loss coefficient. Specifically, we search for the coefficient out of {0, 0.2, 0.4, 0.5, 0.6, 0.8, 1, 2}, and apply the model with the highest CoNLL score on the development set to the test sets.

2.3.3 Results and Discussion

Results of the three approaches (Pipeline, End-to-End and its "no constraint" variant) on the four test sets (LIGHT, AMI, Persuasion, and Switchboard) are shown in Table 3. Specifically, in the "F" columns we show the CoNLL score. To better understand the extent to which the singleton clusters and the non-singleton clusters contribute to overall performance, we report two additional scores. The "ns-F" columns show the CoNLL scores obtained

	LIGHT			AMI			Persuasion			Switchboard		
	F	ns. F	s-F	F	ns-F	s-F	F	ns-F	s-F	F	ns-F	s-F
End-to-End (no constraints)	74.8	55.7	31.5	60.1	43.3	28.0	70.4	51.3	32.2	68.3	51.9	29.1
End-to-End	76.3	57.6	31.5	58.2	43.2	26.1	72.6	54.6	31.8	66.3	51.1	28.2
Pipeline	79.6	60.2	32.9	57.4	42.9	25.2	77.5	56.5	35.3	72.6	53.7	32.5

Table 3: Entity coreference resolution: results of the three approaches on the test sets. For each approach, we show three scores: the unmodified CoNLL score (F), the CoNLL score obtained after removing all the singleton clusters from the system output (ns-F), and the CoNLL score obtained by keeping only the singleton clusters in the system output (s-F).

	MUC			B^3			$CEAF_e$			
	P	R	F	P	R	F	P	R	F	CoNLL
LIGHT	87.8	89.1	88.5	72.7	82.6	77.3	74.6	71.2	72.9	79.6
AMI	66.7	65.5	66.0	48.5	58.3	53.0	59.0	48.3	53.1	57.4
Persuasion	78.7	87.8	83.0	76.6	80.4	78.5	76.4	66.4	71.0	77.5
Switchboard	77.5	79.5	78.5	70.7	74.3	72.4	70.7	63.7	67.0	72.6

Table 4: Entity coreference resolution: official results on the test sets (obtained using the Pipeline approach).

	LIGHT			AMI			Persuasion			Switchboard		
	P	R	F	P	R	F	P	R	F	P	R	F
End-to-End (no constraints)	90.2	88.8	89.5	87.9	81.1	84.4	90.1	87.3	88.7	87.8	82.2	84.9
End-to-End	90.3	88.6	89.5	88.6	76.6	82.2	90.4	86.1	88.2	88.6	79.8	84.0
Pipeline	92.3	91.6	92.0	86.6	78.6	82.4	91.3	89.7	90.5	89.2	86.1	87.6

Table 5: Entity coreference resolution: mention extraction results on the test sets.

by removing the singleton clusters from the output prior to scoring, meaning that the scorers are applied to score only the non-singleton clusters. Similarly, the "s-F" columns show the CoNLL scores obtained by removing the non-singleton clusters from the output prior to scoring, effectively allowing the scorers to score only the singleton clusters.

As we can see, in terms of CoNLL F-score, on LIGHT and Persuasion, Pipeline outperforms the End-to-End, which in turn outperforms its "no constraint" counterpart. A closer look at these results reveals that Pipeline outperforms the other models w.r.t. the identification of both singleton and non-singleton clusters. On Switchboard, Pipeline still offers the best performance, but the use of constraints hurts performance: results on both singleton and non-singleton cluster identification deteriorate. Finally, on AMI we see a completely different trend: the "no constraint" variant offers the best performance while Pipeline performs the worst. While this is somewhat unexpected, recall that Pipeline is trained using setup `T2` while the other models are trained using setup `T1`. We hypothesize that (1) our constraints may not be as universally applicable as we expect, and (2) the distribution of the AMI test set is more similar to the development set in `T1` than that in `T2`, thus allowing better models to be selected.[2] Additional experiments are needed to determine the reason.

Our official test results, expressed in terms of MUC, B^3, and $CEAF_e$ precision (P), recall (R), and F-score (F), are shown in Table 4. While Pipeline underperforms the "no constraint" variant on the AMI test set, our official test set results are all obtained using the Pipeline approach. The reason is that the AMI test set was only released a few days prior to submission, and we did not have enough time to do a systematic comparison of the three models on the AMI test set.

2.3.4 Additional Analysis

Table 5 expresses mention detection results in terms of P, R, and F. Note that (1) these are results obtained on *referring* mentions only; and (2) we consider an entity mention correctly detected if it has an exact match with a gold referring mention in terms of boundary. Comparing the results in Tables 3 and 5, we can see that there is a perfect

[2] We did not have enough time prior to submission to obtain results of all three models using both `T1` and `T2`.

correlation between mention detection results and entity coreference results. These results corroborate results in previous shared tasks on coreference that mention detection plays a crucial role in entity coreference performance and that improving mention detection will likely lead to further improvements in coreference performance.

3 Discourse Deixis Resolution

We cast discourse deixis resolution as identity anaphora resolution. This allows us to continue to use the Xu and Choi (2020) entity coreference model as the baseline. The shared task divides the evaluation of discourse deixis resolution into two phases: (1) the Predicted phase, where a system needs to first identify all of the entity mentions that likely correspond to anaphors and antecedents, then perform discourse deixis resolution on the predicted mentions; and (2) the Gold phase, which is essentially the same as the Predicted phase except that the mentions corresponding to anaphors are to be extracted from the given gold mentions. Below we first describe the system we developed for the Predicted phase and then show how the system we used for the Gold phase differs from the one we used for the Predicted phase. As we will see shortly, our models for both phases perform joint mention extraction and discourse deixis resolution.

3.1 Predicted Phase

As mentioned above, we extend Xu and Choi's (2020) entity coreference model with the following additions, many of which are similar to those we used for entity coreference resolution.

Sentence distance We hypothesize that recency plays a role in resolution, so we add the sentence distance between two spans as a feature.

Type prediction Since the official scorer penalizes a mention e in the system output if e is neither a gold anaphor nor a gold antecedent, we extend the model so that it can predict the type of each span, where the possible types are NULL, ANAPHOR, or ANTECEDENT, and subsequently remove from the output any spans that are predicted to be NULL. For each span x, we predict its type as follows. First, we pass its representation g_x to a FFNN, which outputs a vector ot_x of length 3. Each element $ot_x(t)$ of ot_x denotes the likelihood that span x belongs to type t. The span type t_x is then determined by the type with the highest score.

$$ot_x = \texttt{FFNN}_t(g_x)$$
$$t_x = \arg\max_t ot_x(t)$$

We compute the cross-entropy loss using ot_x. This type loss is then multiplied by a type loss coefficient and added to the loss function of Xu and Choi's model.

Span constraint We enforce the same span constraint that we used for entity coreference, retaining spans in which at most one speaker is involved and whose width is less than 70.

Resolution constraint We enforce a hard constraint on resolution that will be used in both training and inference: only spans corresponding to candidate anaphors can be resolved, and only spans corresponding to candidate antecedents can be chosen as antecedents. We consider a span as a candidate anaphor if it ever appears as an anaphor in the training data, and anything else are considered candidate antecedents.[3]

3.2 Gold Phase

The system we developed for the Gold phase is the same as the one for the Predicted phase except that we modify the span constraint and the resolution constraint, as described below.

Span constraint We retain a span if it corresponds to consecutive sentences that involve only one speaker and is less than 150 words because it is likely to be a candidate antecedent.[4] We also retain a span if it is likely to be a anaphor. Here, we experiment with two heuristics for identifying likely anaphors. Heuristic `CA1` considers any span that corresponds to a gold mention as an anaphor, and Heuristic `CA2` considers any gold mention span that has appeared in the training data as an anaphor.

Resolution constraint We enforce a hard constraint on resolution that will be used in both training and inference: only spans classified in the type prediction step as ANAPHOR can be resolved, and only spans classified in the type prediction step as ANTECEDENT can be selected as antecedents.

[3] We could use the output of type prediction to determine whether a span should be considered as a candidate anaphor or a candidate antecedent, but we did not have time to consider this option during the Predicted phase. We did, however, experiment with this option during the Gold phase.

[4] We use a maximum span width of 150 in the Gold phase because the longest antecedent in the training set is at most 150 words. However, we only used a span width of 70 in the Predicted phase because of memory limitations.

	LIGHT			AMI			Persuasion			Switchboard		
	F	ns-F	s-F	F	ns-F	s-F	F	ns-F	s-F	F	ns-F	s-F
Predicted	**42.7**	47.0	2.2	**35.4**	37.9	2.6	**39.6**	42.1	1.0	**35.4**	39.1	2.0
Gold:T1,CA1,RC+	41.9	43.3	7.2	33.9	32.5	7.5	45.6	45.6	3.8	38.8	36.8	7.4
Gold:T1,CA2,RC+	39.8	40.2	7.3	35.3	34.0	6.4	46.9	47.3	3.2	**40.4**	36.6	9.5
Gold:T1,CA1,RC-	**43.4**	43.9	7.7	31.7	30.5	6.4	46.3	46.2	4.8	40.2	37.2	9.5
Gold:T2,CA1,RC+	38.4	39.8	6.6	**36.9**	35.2	8.4	51.7	53.0	3.0	39.6	36.8	9.4
Gold:T2,CA2,RC+	35.9	34.7	7.8	32.0	28.1	8.6	**52.1**	52.0	4.2	40.2	38.0	8.5
Gold:T2,CA1,RC-	41.6	43.3	6.2	34.7	32.8	5.7	51.2	52.1	3.2	37.9	34.7	8.2

Table 6: Discourse deixis resolution: results of different models on the test sets. For each model, we report three scores: the unmodified CoNLL score (F), the CoNLL score obtained after removing all the singleton clusters from the system output (ns-F), and the CoNLL score obtained by keeping only the singleton clusters in the system output (s-F). The boldfaced scores are our scores on the leaderboard.

3.3 Evaluation

We evaluate the models developed for both the Predicted phase and the Gold phase.

3.3.1 Implementation Details

For all models we use SpanBERT$_{\text{Large}}$ as the encoder. Documents are split into independent segments with a maximum of 512 word pieces, and two segments from each document are used in training. We use different learning rates for BERT-parameters and task-parameters (1×10^{-5} and 3×10^{-4} respectively). Models are trained for 24 epochs with dropout rate 0.3. The type loss coefficient is found using grid search on the test set. See Appendix A for the optimal hyperparameters chosen for each model.

For the Predicted phase, we use method `T1` to create the training and development sets. As in entity coreference, for efficiency reasons the model selects $0.4 \times \#\ of\ words\ in\ the\ document$ spans as top spans for further processing. For each top span, the model selects 50 candidate antecedents for resolution purposes (with the exception of the AMI dataset, where we use only 17 candidate antecedents during inference because of memory limitations).

For the Gold phase, we experiment with six variants. These variants differ along three dimensions: (1) whether `T1` or `T2` is used for partitioning the available annotated data into training and development sets; (2) whether `CA1` or `CA2` is used as the heuristic for identifying likely anaphors; and (3) whether the resolution constraint is used (`RC+`) or not used (`RC-`).

3.3.2 Results and Discussion

Table 6 shows the results of the model used in the Predicted phase as well as the six model variants used in the Gold phase on the test sets. Similar to Table 3, we report three scores for each model: the "F" columns show the CoNLL F-scores; the "ns-F" columns show the CoNLL F-scores obtained by removing the singleton clusters from the system output prior to scoring; and the "s-F" columns show the CoNLL F-scores obtained by removing the non-singleton clusters from the system output prior to scoring. Note that discourse deixis resolution is being viewed as a generalized case of event coreference resolution, and hence the scorer that is used to score entity coreference chains can be used to produce CoNLL F-scores for the output of a discourse deixis resolver.

Our official results for discourse deixis resolution are the boldfaced results in Table 6. Specifically, for the Predicted phase, we submitted the results corresponding to the highest CoNLL score in the first row, and for the Gold phase, we submitted the result of the model variant corresponding to the highest CoNLL score for each test set. The complete set of official results for both phases, which includes the scores obtained via each scorer, is shown in Table 7.

A few points deserve mention. First, while we opted to retain both singleton clusters and non-singleton clusters in the system output, removing the singleton clusters in the Predicted phase can substantially improve performance on all test sets. The reason could be attributed to the fact that singleton cluster identification results are very poor in the Predicted phase, but additional experiments are needed to determine the reason. In contrast, removing singleton clusters from the output in the Gold phase yields worse results in many cases. This could be attributed to the fact that single cluster identification is better in the Gold phase than the Predicted phase, but again additional experiments

	MUC			B^3			$CEAF_e$			
	P	R	F	P	R	F	P	R	F	CoNLL
					Predicted					
Light	44.6	31.2	36.8	56.2	37	44.6	55.3	40.5	46.7	42.7
AMI	45.5	21.2	28.9	52.4	29.5	37.8	44.9	35.1	39.4	35.4
Persuasion	45.5	20.3	28.1	64.9	30.2	41.2	61	41.8	49.6	39.6
Switchboard	35.2	21.3	26.5	52.3	30.4	38.5	50.5	34.9	41.3	35.4
					Gold					
Light	49.0	30.0	37.2	56.3	39.1	46.2	51.7	42.9	46.9	43.4
AMI	44.6	21.2	28.7	49.7	34.6	40.8	39.6	43.0	41.2	36.9
Persuasion	53.3	45.5	49.1	54.9	55.7	55.3	46.0	59.3	51.8	52.1
Switchboard	39.4	31.2	34.8	41.6	48.5	44.8	33.7	55.0	41.8	40.4

Table 7: Discourse deixis resolution: official results on the test sets.

		Light			AMI			Persuasion			Switchboard		
		P	R	F	P	R	F	P	R	F	P	R	F
Anaphor	Predicted	–	73.8	–	–	64.4	–	–	65.9	–	–	71.1	–
	Gold_Best	65.0	65.0	65.0	57.9	61.9	59.8	73.6	77.2	75.4	64.8	74.9	69.5
Antecedent	Predicted	–	27.7	–	–	20.5	–	–	21.2	–	–	21.5	–
	Gold_Best	59.7	33.0	42.5	49.5	32.3	39.1	52.4	58.9	55.5	38.2	52.4	44.2

Table 8: Discourse deixis resolution: mention extraction results on the test sets.

are needed. Second, while all model variants in the Gold phase outperform the model in the Predicted phase on Persuasion and Switchboard, the same is not true on LIGHT and AMI. Specifically, the best models in the Gold phase on these two test sets perform only slightly better than the model in the Predicted phase, and there are many model variants in the Gold phase that underperform the model in the Predicted phase. Finally, the best results for different test sets in the Gold phase are achieved by different model variants. In fact, varying just one of the three dimensions can already trigger non-trivial changes to model performance. Additional experiments are needed to determine the reason.

3.3.3 Additional Analysis

In Table 8, we report the mention extraction results of our officially best models for the Predicted phase and the Gold phase, including both the anaphor extraction results and the antecedent extraction results. Since the best results for different test sets in the Gold phase are achieved by different models, for ease of exposition we refer to them collectively as `Gold_Best`. As in entity coreference, we consider a mention correctly detected in discourse deixis resolution if and only if it has an exact match with a gold mention in terms of boundary.

Two points deserve mention. First, recall from Table 6 that while `Gold_Best` achieves better resolution results than `Predicted` on all four test sets, the differences are substantial on Persuasion and Switchboard but fairly small on LIGHT and AMI. The mention detection results in Table 8 can partially explain why. On Persuasion and Switchboard, `Gold_Best` outperforms `Predicted` for both anaphor detection and antecedent detection. However, on LIGHT and AMI, while `Gold_Best` outperforms `Predicted` for antecedent detection, the reverse is true for anaphor detection. The fact that `Gold_Best` does not consistently outperform `Predicted` for anaphor detection provides suggestive evidence that the Gold setting is not necessarily easier than the Predicted setting for discourse deixis resolution.

Second, `Gold_Best` substantially outperforms `Predicted` for antecedent detection on all four test sets. Two factors could account for this large difference: (1) the maximum span width of `Gold_Best` is a lot larger than that of `Predicted` (150 vs. 70), (2) the candidate antecedents of `Predicted` contain a lot of invalid spans (e.g., a span may start in the middle of a sentence and ends in the middle of another sentence), while the candidate antecedents of `Gold_Best` are restricted to be sentences.

To gain additional insights into the test sets and `Gold_Best`'s performances on them, we show in Table 9 (1) the top five most frequent anaphors in

LIGHT			AMI			Persuasion			Switchboard		
Anaphor	%	F	Anaphor	%	F	Anaphor	%	F	Anaphor	%	F
that	47.5	61.2	that	77.1	42.9	that	64.2	61.8	that	73.8	50.1
it	21.2	40.3	it	8.5	0.0	it	17.9	43.7	it	17.5	15.1
this	11.2	33.5	which	5.9	44.0	this	6.5	64.7	this	1.5	29.8
these terms	1.2	0.0	this	3.4	15.0	the same	3.3	26.1	which	1.1	41.7
more	1.2	0.0	similar effect	0.8	0.0	my research	0.8	0.0	that way	1.1	36.1

Table 9: Discourse deixis resolution: Gold_Best's CoNLL scores on the top five most frequently occurring anaphors.

each test set; (2) the percentage of anaphors that belong to each of the top five anaphors; and (3) `Gold_Best`'s performance on each top anaphor. To measure `Gold_Best`'s performance on each top anaphor, we retain all and only those clusters containing the anaphor in both the gold partition and the system partition and apply the scorer to the resulting partitions.

A few points deserve mention. First, for all test sets, more than 80% of the anaphors are one of the top five most frequently occurring anaphors. As can be seen, "that" and "it" are the most frequent anaphors in all test sets, followed by "this" and "which". As for resolution of the most frequent anaphors, the results for "that" are consistently among the best, and the same can be said for "which". However, the resolution results of "this" and "it" are comparatively less consistent across datasets. For example, the resolution of "it" appears to be much better on LIGHT and Persuasion than on AMI and Switchboard.

4 Bridging Resolution

The shared task divides the evaluation of bridging anaphora resolution into two phases: (1) the Predicted phase, where a system needs to first identify all of the entity mentions that likely correspond to anaphors and antecedents, then perform bridging resolution on the predicted mentions; and (2) the Gold phase, which is essentially the same as the Predicted phase except that bridging resolution is performed on the given gold mentions. Below we describe our approach and our official test results.

4.1 Approach

We employ a multi-pass sieve approach to bridging resolution. Our decision to employ a sieve-based approach is motivated in part by its successful application to bridging resolution in our previous work (Kobayashi and Ng, 2021), where we achieved state-of-the-art results by applying a rule-based sieve followed by a learning-based sieve.

The multi-pass sieve approach to entity coreference resolution, which was originally proposed by members of the Stanford NLP Group (Raghunathan et al., 2010), received a lot of attention in the coreference research community after their team won the CoNLL 2011 shared task on Unrestricted Coreference Resolution (Lee et al., 2011). Briefly, a *sieve* is composed of one or more heuristic rules. When applied to entity coreference resolution, each rule extracts a coreference relation between two mentions based on one or more conditions. For example, one rule in Stanford's discourse processing sieve posits two mentions as coreferent if they are both pronouns and are produced by the same speaker. Sieves are ordered by their precision, with the most precise sieve appearing first. To resolve a set of mentions in a document, the resolver makes multiple passes over them: in the i-th pass, it attempts to use only the rules in the i-th sieve to find an antecedent for each mention m_k. Specifically, when searching for an antecedent for m_k, its candidate antecedents are visited in an order determined by their positions in the associated parse tree (Haghighi and Klein, 2009). The partial clustering of the mentions created in the i-th pass is then passed to the $i+1$-th pass. Hence, later passes can exploit the information computed by previous passes, but a coreference link established earlier cannot be removed later.

Our sieve-based approach to bridging resolution differs from the conventional approach described above in two key aspects. First, rather than order the sieves by precision, we order them so that they collectively achieve the best performance on the test sets. Second, later sieves only attempt to resolve mentions that have not been resolved by earlier sieves (i.e., earlier decisions will not be overridden), but otherwise do not exploit the information computed by earlier sieves. Below we first present our sieves and then describe how we

perform mention extraction.

4.1.1 Sieves

We employ five learning-based sieves, including one neural sieve and four "same head" sieves.

4.1.1.1 Neural Sieve

Our neural sieve uses Yu and Poesio's (2020) multi-task learning (MTL) based neural bridging resolver, which has achieved state-of-the-art results on standard evaluation corpora for bridging resolution.[5] Yu and Poesio presented two extensions to Kantor and Globerson's (2019) span-based neural entity coreference model. First, they provided gold mentions as input to the model, meaning that the model needs to learn the span representations but not the span boundaries. Second, they proposed to train the model to perform coreference and bridging in a MTL framework, where the span representation layer is shared by the two tasks so that information learned from one task can be utilized when learning the other task. Unlike feature-based approaches, where feature engineering plays a critical role in performance, this model employs only two features, the length of a mention and the mention-pair distance. To adapt the model to the dialogue domain, we have also added a feature that encodes the turn distance between mentions, where a turn is defined as a set of contiguous sentences by the same speaker.

In preliminary experiments with the MTL model, we found that resolution recall and precision can sometimes be fairly imbalanced. We hypothesize that having more balanced recall and precision numbers could result in better resolution F-score. To make recall-precision tradeoffs, we tune the *dummy score*, which is a mention-pair score assigned by the MTL model to the dummy candidate antecedent of a candidate anaphor and set to 0 by default. To understand why tuning the dummy score allows us to make resolution recall-precision tradeoffs, recall that the score reflects how likely the corresponding candidate antecedent will be chosen as the antecedent for the candidate anaphor under consideration. Hence, a higher dummy score makes it more likely for the anaphoric candidate to be resolved to the dummy antecedent, thereby potentially reducing recall and possibly improving precision. In contrast, a lower dummy score makes it less likely for the anaphoric candidate to be resolved to the dummy antecedent, thereby potentially reducing precision and improving recall.

4.1.1.2 Same-Head Sieves

Next, we design four sieves, all of which focus on establishing bridging links between two mentions that have the same head lemma. We therefore refer to them as *same-head* sieves. We focus on same-head sieves as opposed to different-head sieves because the former are arguably less challenging to design than the latter, especially in the Predicted setting where no gold mentions are given.

Each sieve operates by combining a *mention-pair model* (Soon et al., 2001; Ng and Cardie, 2002), which in our case is a binary classifier that determines whether two mentions having the same head are involved in a bridging relation, with a *closest-first single-link clustering* algorithm, which selects as the antecedent of an anaphoric candidate the closest preceding mention that is classified as its bridging antecedent. Motivated in part by the same-head bridging resolution rules developed by Rösiger (2018), we divide the same-head bridging links into four groups based on whether the anaphor, m_i, is singular or plural and whether the candidate antecedent, m_j, is singular or plural, and create one sieve for each group of bridging links. More specifically, the four sieves are: singular-singular (both mentions are singular), singular-plural (m_i is singular and m_j is plural), plural-singular (m_i is plural and m_j is singular), and plural-plural (both mentions are plural).

To train the mention-pair models, we use the SVM learner implemented in the SVM^{light} software package (Joachims, 1998). The training instances for a sieve include those that (1) correspond to same-head pairs and (2) match the singularity/plurality conditions on the anaphor and the candidate antecedents for the sieve under consideration. For instance, the training instances for the singular-singular sieve include only those training instances for which the mentions have the same head and are both singular. Positive training instances are created by pairing each anaphoric candidate with a gold antecedent or preceding mentions that are coreferent with the gold antecedent, following Yu and Poesio (2020). Negative training instances are created by pairing anaphoric mentions with preceding mentions that are not correct antecedents or by pairing non-anaphoric mentions with preced-

[5] We use their publicly available implementation from https://github.com/juntaoy/dali-bridging. All model parameters are set to the same values as in Yu and Poesio (2020) except for training, which will be described in Section 4.2.

ID	Features	Description	Motivation	Used by S-S	S-P	P-S	P-P
1	premodifiers' POS tags	the POS tag of the premodifier of m_i's head, if any, and the POS tag of the premodifier of m_j's head, if any; only three POS tags are considered: JJ, NN, VBN	presence of certain types of premodifiers may influence bridging decisions	✓	✓	✓	✓
2	speaker	whether the speakers of m_i and m_j are the same	may influence bridging decisions	✓	✓	✓	✓
3	turn distance	the turn distance between m_i and m_j binned into the following buckets: 1, 2, 3, 4, 5−7, 8−15, 16−31, 32−63, 64+	bridging link is less likely as distance increases	✓	✓	✓	✓
4	conflicting article pairs	features that encode the presence of select unordered pairs of articles possessed by m_i and m_j (see Table 11 for the list)	bridging link is more likely if the mentions possess any of these conflicting pairs	✓			
5	the-anaphor	whether m_i starts with "the"	could improve resolution precision			✓	
6	bare noun/article pairs	features that encode whether one mention is a bare noun (phrase) and the other possesses one of the select articles[6]	bridging link is more likely if any of these features is fired			✓	
7	dataset	the name of the dataset (e.g., AMI) in which m_i and m_j appears	could affect bridging decisions	✓		✓	

Table 10: Bridging resolution: features for training the four same-head sieves, namely, the singular-singular (S-S) sieve, the singular-plural (S-P) sieve, the plural-singular (P-S) sieve, and the plural-plural (P-P) sieve. Each feature encodes the candidate anaphor, m_i, and a candidate antecedent, m_j, or the relationship between them. The presence of a checkmark in row x and column y indicates that feature x is used to train sieve y.

ing mentions. We employ a *turn window* when generating training instances, meaning that training instances will be generated from two mentions only if the distance between them is within a certain number of turns. We treat the turn window as a tunable parameter. Test instances are created in the same manner as the training instances.

The four mention-pair models are trained using different features. Table 10 provides a description of these features, the motivation behind their design, as well as the features used to train each mention-pair model. We use Stanford CoreNLP (Manning et al., 2014) and spaCy (Honnibal et al., 2020) to extract the linguistic information needed to compute the features. A few features deserve mention. Feature 4, which is used by the singular-singular sieve, is a set of features encoding the presence of *conflicting* article pairs, which are manually identified via our inspection of the official development sets. Most conflicting pairs are composed of a definite article and an indefinite article, so the presence of a conflicting article pair makes the corresponding mentions less likely to be coreferent (because they differ in definiteness). At the same time, however, the presence of a conflicting pair makes the corresponding mentions more likely to have a bridging relation. The same can be said for feature 6. Specifically, this group of features is applicable when one mention is a bare expression (which indicates genericity) and the other is a definite expression, and hence their presence makes it more likely for the two mentions to have a bridging relation.

[6]The select articles are: some, another, the, these, those, many, most, and all of the possessive personal pronouns.

the/a, the/the other, the/another, one/the, one/the other, an/another, that/their, the/your, my/your, an/the, that/next, the/next, my/a, the/your, the/my, your/my, my/this, a/your, the/some, this/next, a/our, such a/the, a/your, some/the, the/more, a/one, a/this, an/her, your/'s, an/[adj.] bare, a/[adj.] bare, [adj.] bare/the, any/[adj.] bare, some/[adj.] bare, this/[adj.] bare, our/[adj.] bare, my/[adj.] bare, some/[adj.] bare

Table 11: Conflicting article pairs. [adj.] bare means that a mention is a bare expression that is optionally premodified by an adjective.

4.1.2 Predicted vs. Gold Phases

The systems we developed for the Gold phase and the Predicted phase differ primarily in mention extraction. For the Predicted phase, the mentions used by the neural sieve and the same-head sieves are extracted differently. For the neural sieve, since Yu and Poesio's (2020) MTL model assumes as input the gold mentions in the input document, we modify their model to enable automatic learning of

span boundaries. For the same-head sieves, we use the noun phrases heuristically extracted from syntactic parse trees using Stanford CoreNLP (Manning et al., 2014). For the Gold phase, anaphors and their candidate antecedents are restricted to be gold mentions. More specifically, during training, we use the given gold mentions to train the neural sieve and the same-head sieves, and during inference, we only resolve those gold mentions that are predicted to be bridging anaphors. The model that predicts whether a gold mention is a bridging anaphor is trained by removing the FFNN layers for the bridging task and the coreference task from Yu and Poesio's (2020) MTL model and changing the loss function to sigmoid cross entropy, which is commonly used for binary classification tasks.

4.2 Evaluation

Next, we evaluate our sieve-based approach to bridging resolution.

4.2.1 Model Training and Parameter Tuning

In this subsection, we describe the data we use to (pre)train our models (i.e., the MTL model, the four SVM models, and the anaphor detection model) and the parameters we tune for these models.

The MTL model To train the MTL model, we first pre-train a model on the non-dialogue datasets, including ARRAU RST (train, dev, and test), Gnome, and Pear, and then fine-tune the resulting model on the dialogue datasets, including Trains 93 and the dev sets for Trains91, LIGHT, AMI, Persuasion, and Switchboard. Pre-training takes 15000 steps and fine-tuning takes an additional 7000 steps for both the Predicted and Gold settings.

The only parameter we tune for the MTL model is the dummy score, which we use to make precision-recall tradeoffs, as mentioned before. We tried dummy scores of 0, 0.5, 1.0, 1.5, and 2.0 and selected the score such that the neural sieve alone achieved the highest resolution F-score on the test sets.[7] Note that selecting the default dummy score (i.e., 0) is equivalent to not making any adjustment to the recall and precision scores produced by the MTL model.

The SVM models We train the SVM models using a linear kernel on the dev sets of LIGHT, AMI, Persuasion, and Switchboard. We tune two parameters. The first parameter is the regularization parameter C, which we search out of {1, 10, 100, 1000}. The second parameter is the turn window size. We consider window sizes from 2 to 10. Both parameters are tuned to maximize resolution F-score on the same data on which the models are trained.

Sieve ordering and removal Another parameter we tune involves sieve application. In particular, we need to determine (1) the order in which the sieves should be applied and (2) which SVM sieves should be retained/removed (we take the neural sieve as the basic sieve and do not consider removing it). The ordering that we select is the one that yields the highest resolution F-score on the test sets. Owing to time limitations, it is not feasible to exhaustively try all possible orderings, so we experiment with the following orderings. For the Predicted setting, we first determine whether we should apply all the SVM sieves before or after the neural sieve, then try to order the SVM sieves, and finally consider removing certain SVM sieves. For the Gold setting, we first experiment with as many ways of ordering the neural sieve and the SVM sieves as time permits, and then consider removing certain SVM sieves.

The anaphor detection model We pre-train and fine-tune the anaphor detection model in the same way as we pre-train and fine-tune the neural sieve. Pre-training takes 2000 steps and fine-tuning takes an additional 2000 steps. The only parameter we tune is the anaphoric fraction (AF), which we search out of {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}. Specifically, we select the AF that yields the highest resolution F-score on the test sets. Note that the tuning of AF is done after the sieves are selected and ordered.

Details of the selected hyperparameters can be found in Appendix A.

4.2.2 Results and Discussion

Table 12 shows the recognition and resolution results of our sieve-based approach to bridging resolution on the test sets. These are also our official results for the bridging track. For the Predicted setting, we achieve resolution F-scores of 13.3–21.9%. For the Gold setting, we achieve resolution F-scores of 19.6–31.4%. Note that these are entity-based F-scores, meaning that a bridging anaphor is considered correctly resolved as long as it is resolved to its antecedent or a preceding mention that is coreferent with its antecedent. As can be seen, the system performs better in the Gold setting with

[7] We could have tuned this parameter so that it maximized the resolution F-score achieved by the system rather than just the neural sieve, but could not do that due to time limitations.

		LIGHT			AMI			Persuasion			Switchboard		
		P	R	F	P	R	F	P	R	F	P	R	F
Predicted	Recognition	21.8	44.3	29.2	26.8	32.9	29.5	29.6	38.9	33.6	28.9	32.2	30.4
	Resolution	10.4	21.2	14.0	12.1	14.8	13.3	19.3	25.3	21.9	14.5	16.1	15.3
Gold	Recognition	34.7	40.7	37.5	37.0	42.2	39.4	43.0	52.1	47.1	37.7	50.9	43.3
	Resolution	18.3	21.4	19.7	18.4	21.0	19.6	28.7	34.7	31.4	18.4	24.8	21.1

Table 12: Bridging resolution: recognition and resolution results of our sieve-based approach on the test sets. These are also our official test set results.

respect to both recognition and resolution. These results are consistent with our intuition that the Gold setting is easier than the Predicted setting.

4.2.3 Additional Analysis

To gain additional insights into our approach, we perform two experiments. Since we did not store the parameters of the model that produced the official test results, we had to retrain the model to produce the results in these experiments. Hence, the performance numbers in these experiments may not be directly comparable to the official test results.

In the first experiment, we examine the contribution of each sieve to the overall performance of our approach. The recognition and resolution F-scores of this experiment are shown in Table 13. As can be seen, the optimal sieve ordering is determined to be the singular-plural sieve, followed by the plural-plural sieve, the plural-singular sieve, and MTL, and finally the singular-singular sieve. The sieve removal process begins after the optimal sieve ordering is determined. While sieve ordering is dataset-independent, sieve removal is performed in a dataset-dependent manner. As shown in the table, a checkmark appears next to a sieve if and only if the corresponding sieve is retained. For the most part, we can see that each sieve contributes positively to overall performance. Hence, these results demonstrate the usefulness of a sieve-based approach to bridging resolution.

In the second experiment, we examine the effects of (1) anaphor detection and (2) recall-precision balancing in the neural sieve on the overall performance of our approach by *ablating* one or both of these components/factors in our approach. Note that anaphor detection is applicable in the Gold setting only. The results of this experiment are shown in Table 14. For comparison purposes, we show the results achieved by MTL, the results of our approach (in "Ours") and the different ablated versions of our approach, including "Ours w/o AD" (our approach with anaphor detection ablated), "Ours w/o B" (our approach without recall-precision balancing), and "Ours w/o B&AD" (our approach with both of them ablated). As we can see, while the unablated version of our approach does not achieve the highest recognition F-score on all datasets, it does achieve the highest resolution F-score on all of them. A closer examination of the ablated results in the Gold setting reveals that the best resolution F-scores can sometimes be achieved without anaphor detection or recall-precision balancing, but applying them in combination yields the best result on all datasets.

5 Conclusions

We presented our systems we developed for the three tracks of the CODI-CRAC 2021 shared task, namely entity coreference resolution, discourse deixis resolution, and bridging resolution. Our team ranked second for entity coreference resolution, first for discourse deixis resolution, and first for bridging resolution. For entity coreference resolution, our analysis of our three approaches (Pipeline, End-to-End, as well as its "no constraint" variant) revealed that there is a perfect correlation between entity coreference performance and mention detection, suggesting that coreference performance can further be improved by improving mention detection. For discourse deixis resolution, our analysis revealed that (1) contrary to common wisdom, anaphor detection was not always easier in the Gold setting (where gold mentions were given) than the Predicted setting; and (2) substantial gains could be achieved simply by removing the singleton clusters from the output prior to scoring, especially in the Predicted setting. For bridging resolution, our results suggested that the Gold setting is easier than the Predicted setting, and our analysis showed that the sieves, anaphor detection, and recall-precision balancing all contributed positively to overall system performance. Finally, for a fur-

	LIGHT			AMI			Persuasion			Switchboard		
	Sieve	Recog	Resol	Sieve	Recog	Resol	Sieve	Recog	Resol	Sieve	Recog	Resol
Predicted												
Sing-plur	✓	1.4	0.9	✓	3.3	1.5		–	–		–	–
+ Plur-plur	✓	2.2	1.3		–	–		–	–		–	–
+ Plur-sing		–	–		–	–		–	–		–	–
+ MTL	✓	27.9	13.2	✓	26.0	12.5	✓	33.9	21.0	✓	26.0	13.5
+ Sing-sing		–	–	✓	26.3	12.7	✓	34.7	21.2		–	–
Gold												
Sing-plur	✓	1.4	1.0	✓	2.8	1.8	✓	3.4	2.0	✓	1.7	1.0
+ Plur-plur	✓	2.8	1.9		–	–		–	–		–	–
+ Plur-sing	✓	3.7	2.3	✓	6.0	3.3		–	–	✓	2.5	1.5
+ MTL	✓	38.2	21.4	✓	37.2	18.3	✓	47.7	32.5	✓	44.6	21.5
+ Sing-sing		–	–		–	–		–	–	✓	44.6	21.6

Table 13: Bridging resolution: recognition and resolution F-scores of our sieve-based approach on the test sets using the optimal sieve ordering and combination determined on development data. The models used in these experiments are (re)trained after the official submission, so these performance numbers may not be directly comparable to our official test set results.

		LIGHT			AMI			Persuasion			Switchboard		
		P	R	F	P	R	F	P	R	F	P	R	F
Predicted													
Recognition	MTL	22.9	36.1	**28.0**	23.5	32.2	27.2	24.0	56.3	33.7	22.4	45.1	**29.9**
	Ours w/o B	22.5	36.9	27.9	23.5	33.0	**27.4**	23.8	57.6	33.7	22.4	45.1	**29.9**
	Ours	22.5	36.9	27.9	25.8	26.9	26.3	29.6	42.0	**34.7**	27.5	24.7	26.0
Resolution	MTL	10.7	16.9	13.1	10.0	13.7	11.6	14.1	33.0	19.8	9.8	19.7	13.0
	Ours w/o B	10.6	17.3	**13.2**	10.1	14.2	11.8	13.9	33.7	19.7	9.8	19.7	13.0
	Ours	10.6	17.3	**13.2**	12.4	13.0	**12.7**	18.1	25.7	**21.2**	14.2	12.8	**13.5**
Gold													
Recognition	MTL	34.5	50.8	41.1	33.7	37.8	35.6	35.5	63.5	45.5	35.2	53.8	42.6
	Ours w/o B&AD	34.6	51.3	41.4	33.6	38.1	35.7	35.5	63.9	45.6	35.0	54.2	42.5
	Ours w/o B	35.9	49.2	**41.5**	38.0	36.3	**37.2**	36.7	62.5	46.2	39.0	52.1	**44.6**
	Ours w/o AD	39.7	37.1	38.4	33.6	38.1	35.7	41.4	54.9	47.2	35.0	54.2	42.5
	Ours	40.4	36.1	38.2	38.0	36.3	**37.2**	42.4	54.5	**47.7**	39.0	52.1	**44.6**
Resolution	MTL	16.7	24.6	19.9	16.4	18.3	17.3	23.3	41.7	29.9	16.9	25.8	20.4
	Ours w/o B&AD	17.2	25.5	20.6	16.6	18.8	17.6	23.3	42.0	30.0	16.9	26.2	20.6
	Ours w/o B	18.0	24.6	20.8	18.7	17.9	**18.3**	24.2	41.3	30.6	18.9	25.2	**21.6**
	Ours w/o AD	22.2	20.7	**21.4**	16.6	18.8	17.6	28.0	37.2	31.9	16.9	26.2	20.6
	Ours	22.6	20.2	**21.4**	18.7	17.9	**18.3**	28.9	37.2	**32.5**	18.9	25.2	**21.6**

Table 14: Bridging resolution: effect of anaphor detection and recall-precision balancing on the recognition and resolution performance of our sieve-based approach on the test sets. The models used in these experiments are (re)trained after the official submission, so these performance numbers may not be directly comparable to our official test set results.

ther analysis of our systems and other participating systems, we refer the reader to our cross-team analysis paper (Li et al., 2021), which also contains a discussion of the lessons we learned and our vision of how the field should move forward.

Acknowledgments

This work was supported in part by NSF Grant IIS-1528037. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the NSF.

References

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161, Singapore. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Zenodo.

Thorsten Joachims. 1998. Making large-scale support vector machine learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexan-

der J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT press, Cambridge, USA.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

Sopan Khosla, Juntao Yu, Ramesh Manuvinakurike, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé. 2021. The CODI-CRAC 2021 shared task on anaphora, bridging, and discourse deixis in dialogue. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*. Association for Computational Linguistics.

Hideo Kobayashi and Vincent Ng. 2021. Bridging resolution: Making sense of the state of the art. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1652–1659, Online. Association for Computational Linguistics.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA. 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. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Shengjie Li, Hideo Kobayashi, and Vincent Ng. 2021. The CODI-CRAC 2021 shared task on anaphora, bridging, and discourse deixis resolution in dialogue: A cross-team analysis. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*. Association for Computational Linguistics.

Jing Lu and Vincent Ng. 2020. Conundrums in entity coreference resolution: Making sense of the state of the art. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6620–6631, Online. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora resolution with the ARRAU corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 11–22, New Orleans, Louisiana. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501.

Ina Rösiger. 2018. Rule- and learning-based methods for bridging resolution in the ARRAU corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 23–33.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Juntao Yu and Massimo Poesio. 2020. Multitask learning-based neural bridging reference resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3534–3546, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Model Hyperparameters

Entity coreference:

- End-to-end (with and without constraints): the model is trained using a type loss coefficient of 0.2.
- Pipeline: the type prediction model uses a type loss coefficient of 500 and the coreference model uses a type loss coefficient of 0.2.

Discourse deixis:

- Predicted: for Persuasion, the type loss is 0.2; for other datasets, the type loss is 0.5.
- Gold:T1,CA1,RC+: for all datasets, the type loss is 0.2.
- Gold:T1,CA2,RC+: for Persuasion and AMI, the type loss is 0.2; for Light and Switchboard, the type loss is 0.5.
- Gold:T1,CA1,RC-: for all datasets, the type loss is 0.2.
- Gold:T2,CA1,RC+: for all datasets, the type loss is 0.2.
- Gold:T2,CA2,RC+: for Switchboard, the type loss is 0.2; for other datasets, type loss is 0.5.
- Gold:T2,CA2,RC-: for Persuasion, the type loss is 0.5; for other datasets, the type loss is 0.2.

Bridging:

- Neural sieve: For the Predicted setting, the selected dummy scores are: 0.0 for LIGHT, 0.5 for AMI, 2.0 for Persuasion, and 1.5 for Switchboard. For the Gold setting, the scores are: 1.0 for LIGHT, 0.0 for AMI, 1.0 for Persuasion, and 0.0 for Switchboard.
- SVM sieves: For the singular-singular sieve, the turn window sizes are 8 for the Predicted setting and 4 for the Gold setting. For each of the remaining sieves, the turn window size is the same for both settings, namely 10 for the singular-plural sieve, 5 for the plural-singular sieve, and 6 for the plural-plural sieve. The SVM regularization parameter is 100 for all SVM models.
- Optimal sieve ordering for each setting and the sieves selected for each dataset ("Lgt", "Prssn", and "Swbd" correspond to LIGHT, Persuasion, and Switchboard respectively):

	Lgt	AMI	Prssn	Swbd
		Predicted		
Sing-plur	✓	✓		
+ Plur-plur	✓			
+ Plur-sing				
+ MTL	✓	✓	✓	✓
+ Sing-sing		✓	✓	
		Gold		
Sing-plur	✓	✓	✓	✓
+ Plur-plur	✓			
+ Plur-sing	✓	✓		✓
+ MTL	✓	✓	✓	✓
+ Sing-sing				✓

- Anaphor detection model: The selected ratio is 0.4.