The Pipeline Model for Resolution of Anaphoric Reference and Resolution of Entity Reference

Hongjin Kim[*], **Damrin Kim**[*], **Harksoo Kim**
Konkuk University / Seoul, South Korea
{jin3430, ekafls33, nlpdrkim}@konkuk.ac.kr

Abstract

The objective of anaphora resolution in dialogue shared-task is to go above and beyond the simple cases of coreference resolution in written text on which NLP has mostly focused so far, which arguably overestimate the performance of current SOTA models. The anaphora resolution in dialogue shared-task consists of three subtasks; subtask1, resolution of anaphoric identity and non-referring expression identification, subtask2, resolution of bridging references, and subtask3, resolution of discourse deixis/abstract anaphora. In this paper, we propose the pipelined model (i.e., a resolution of anaphoric identity and a resolution of bridging references) for the subtask1 and the subtask2. In the subtask1, our model detects mention via the parentheses prediction. Then, we create mention representation using the token representation constituting the mention. Mention representation is fed to the coreference resolution model for clustering. In the subtask2, our model resolves bridging references via a MRC framework. We construct a query for each entity with *"What is related of ENTITY?"*. The input of our model is query and documents(i.e., all utterances of dialogue). Then, our model predicts entity span that is answer for a query.

1 Introduction

Coreference Resolution (CR) is a task which identifying and clustering mentions referring to the same entity in text. Multiple benchmark datasets of CR have been developed in recent years, However, most of these corpora only focus on identity coreference resolution and neglect relations like discourse deixis or bridging anaphora. The goal of the anaphora resolution in dialogue shared-task is to go above and beyond the simple cases of coreference resolution in these datasets (Khosla et al., 2021). This shared-task consists of three subtasks; subtask1 for resolution of anaphoric identity and non-referring expression identification, subtask2 for identifying references to entities related to already introduced entities, and subtask3 for resolution of discourse deixis/abstract anaphora. The dataset consists conversations, so it might has grammatical error. To deal with this dataset, we need to perform speaker grounding of pronouns and focus on long-distance conversation structure. In this paper, we propose the pipelined model for the subtask1 and the subtask2 (i.e., a resolution of anaphoric identity and a resolution of bridging references).

2 Task Description

Track	**AR**	**BR**	
		Pred	**Gold**
System	Sec. 3.1	Sec. 3.2	
Setting	Pred	Pred	Gold
Baselines	-	-	
Learning framework	Pointer Net.	MRC	
Markable identification	Sec. 3.1.1	Sec. 3.1.1	
Train. data	STD	QUOREF, STD	
Dev. data	STD	STD	

Table 1: System summary

In the subtask1, our goal is to identify coreference chains which consist more cases of including split antecedent and singleton. We propose two

[*] equal contribution

Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue, pages 43–47
Punta Cana, Dominican Republic, November 10, 2021. ©2021 Association for Computational Linguistics

models for the subtask1, First is the mention detection model and second is the coreference resoultion model. Our mention detection model identifies mention via the parentheses prediction. Then, based on the prediction of mention detection model, we create the mention representation using the token representation constituting the mention. Finally, mention representation is fed to the coreference resolution model for clustering. In the subtask2, our goal is to identify associative anaphoric references (i.e., references to entities related to already introduced entities). Inspired by Li et al. (2019), we propose a machine reading comprehension (MRC) framework for the subtask2. Given a question, MRC models predict answer spans from a context. Following this framework, we transformed the subtask2 to a QA task: each bridging pair $P(e_i, e_j)$ can be reconstructed as a question $q(x)$ and a answer y. We constructed the question to *"What is related of x?"*. The input of our bridging reference model is a question and the document(i.e., all utterances of the dialogue). Then, our model extracts the related entity(y) span that is an answer for the question. Our MRC model is pre-trained using QUOREF Dasigi et al. (2019) dataset. Our systems were developed using the shared-task datasets (STD) and evaluated in the tracks Eval-AR, Eval-Br(Pred) and Eval-Br(Gold), respectivly. See Table 1 for a summary.

3 Key Components of Our Model

3.1 Subtask1

3.1.1 Mention Detection

In mention detection, we perform mention detection via parentheses prediction for the token of each utterance. All mentions in the dataset is indicated in parentheses and some of these are the nested mentions. Since some of mentions are nested, traditional sequence labeling is not fit for the mention detection. For example, In *"((Health) and (education))"*, it can be expressed to [B-I-I(*Health and education*), B-O-O(*Health*), O-O-B(*education*)] by the BIO tag. To address this issue, we propose a method to predict the opening parentheses and the closing parentheses independently. Firstly we need to transform the tagging-style of the dataset. For the opening parentheses, *"((Health) and (education))"* can be tagged to ["((", "O", "("]. For the closing parentheses, it can be tagged to [")", "O", "))"].

We adopt pre-trained ELECTRA-large model Clark et al. (2020) as a backbone. Given an each utterance, $U = \{u_1, u_2, ..., u_n\}$, where n denotes the number of tokens, ELECTRA receives the input tokens and outputs the context token representation $E = \{E_1, E_2, ..., E_n\}$. For the parentheses prediction layer, we adopt the label attention network (LAN) Cui and Zhang (2019). The context token representation E is used as input to the parentheses prediction layer which is consists the opening parentheses prediction and the closing parentheses prediction. The parentheses prediction layer calculates the association of the token representation with the opening and closing parenthesis embedding vectors, the opening and closing parenthesis predictions are performed independently. More detail, the token representation E is fed to BILSTM as follows:

$$\begin{aligned} \overleftrightarrow{h}^1_i &= \text{LSTM}(E_i, \overleftrightarrow{h}^1_{i-1}) \\ \overleftrightarrow{H}^1 &= \{\overleftrightarrow{h}^1_1, \overleftrightarrow{h}^1_2, ..., \overleftrightarrow{h}^1_n\} \end{aligned} \quad (1)$$

For the label attention, the scaled dot-product attention produces an attention matrix α consisting of a potential label distribution for each token. We define $Q = \overleftrightarrow{H}^1, K = V = Emb_k$, where k denotes the number of the parentheses. In embedding vector, two embedding vector tables are randomly initialized. One is the opening parentheses embedding and another is the closing parentheses embedding. Opening and closing parenthesis embedding are used in opening and closing parenthesis predictions, respectively. In other words, Emb_k can be Emb_k^{open} or Emb_k^{close}. The label attention vector is calculated as follows:

$$\begin{aligned} A^1(C_i) &= attention(Q, K, V) = \alpha V \\ \alpha &= softmax(\frac{Q * K^T}{\sqrt{d_h}}) \end{aligned} \quad (2)$$

Then, we concatenate the hidden states of BILSTM $\overleftrightarrow{H}^1$ and the label attention vector $A^1(C_i)$ to represent $H^1 = [\overleftrightarrow{H}^1; A^1(C_i)]$. H^1 is fed to subsequent BILSTM as follows:

$$\begin{aligned} \overleftrightarrow{h}^2_i &= \text{LSTM}(H^1, \overleftrightarrow{h}^2_{i-1}) \\ \overleftrightarrow{H}^2 &= \{\overleftrightarrow{h}^2_1, \overleftrightarrow{h}^2_2, ..., \overleftrightarrow{h}^2_n\} \end{aligned} \quad (3)$$

Finally, the parentheses of each token is predicted based on the attention scores as follows:

$$\begin{aligned} A^2(C_i) &= softmax(\frac{Q * K^T}{\sqrt{d_h}}) \\ \hat{y}_i^{opening} &= argmax(A^2(C_i^{opening})) \\ \hat{y}_i^{closing} &= argmax(A^2(C_i^{closing})) \end{aligned} \quad (4)$$

We define $Q = \overleftrightarrow{H}^2, K = Emb_k$. It trains to minimize the cross-entropy between the predicted parenthesis and the correct parenthesis as follows:

$$
\begin{aligned}
loss_{opening} &= -\sum_i \hat{y}_i^{opening} \log y^{opening}{}_i \\
loss_{closing} &= -\sum_i \hat{y}_i^{closing} \log y^{closing}{}_i \\
loss_{total} &= 0.5 * loss_{opening} + 0.5 * loss_{closing}
\end{aligned}
\tag{5}
$$

However, the parenthesis prediction layer may predict the number of opening and closing parentheses differently. In this case, we extract mentions based on the number of opening parentheses without using any post-processing.

3.1.2 Coreference Resolution

In coreference resolution, we create the mention representation using the token representation constituting the mention as follows:

$$
M_i = \frac{1}{N-k+1} \sum_k^N \overleftrightarrow{h}_k^1 \tag{6}
$$

$\overleftrightarrow{h}_k^1$ is the contextual token representation in the Equation 1. Then, the mention representations $M = \{M_1, M_2, ...M_l\}$, where l denotes the number of mentions, are fed to coreference model. We also use ELECTRA which is fine-tuned in the mention detection model. To resolve coreference, all mentions in the document need to be fed to system. However, due to the restriction of the input length, all mentions can not be fed to model at once. Therefore, we segment according to the maximum input length. Using the pointer network Vinyals et al. (2015), our model is trained to point to the mention that is referencing the closest distance. For example, if three mentions (M_i, M_j, M_k) are in the same cluster, our model is trained that (M_i, M_j, M_k) points to $(M_j, M_k, root)$, respectively (assume $i < j < k$ and mentions are sorted by the order of appearance). In case of the singleton, Our model is trained that the singleton points to itself. In the pointer network, the self-attention mechanism is used to point the mention that is referencing as follows:

$$
\begin{aligned}
A(M_i) &= \frac{M * M}{\sqrt{d_h}} \\
\hat{y}_i^{coref} &= argmax(softmax(A(M_i)))
\end{aligned}
\tag{7}
$$

For clustering, we need to post-process the prediction of our model. Since our model is trained to predict to the closest reference, we merge the cluster sequentially. If M_i points to M_j, we merge these mentions as $[M_i, M_j]$ and if M_j points to the next mention, it will be merged sequentially. If M_i points to M_i (i.e., singleton), we create the new cluster $[M_i]$. We only use this post-processing method without any other methods.

3.2 Subtask2

In the bridging references, we propose a MRC framework. We adopt pre-trained RoBERTa-large model Liu et al. (2019) as a backbone. In our MRC model, when given a query *"What is related of x"*, each token representation in document that is the output of the RoBERTa-large model is fed into a fully connected layer for predicting a starting position and an ending position of the related entity span. Since our MRC model answers for a query, it is important to decide which word to substitute for x in a query. The MRC model can be confused by the pronouns in a query, since a document might have many pronouns of the same word. We assume that the first mention (i.e., the earliest appeared mention) is not a pronoun but an ANCHOR. To avoid x being replaced by a pronoun, we substitute x with the first mention in each cluster that is result of the subtask1. The cluster $C = \{c_1, c_2, ...c_n\}$, where n denotes the number of clusters, is the result of our coreference resolution model. The mention $M_{i,1}$ in each cluster $c_i = \{M_{i,1}, M_{i,2}, ..., M_{i,j}\}$ is used as "x". j denotes the number of mentions in each cluster. For example, If the mention $M_{i,1}$ references $M_{k,1}$, a query is "what is related of $M_{i,1}$?" and our system predicts the span of $M_{k,1}$. The input of our model is a query and the document (i.e., all utterances) as *"[CLS] query [SEP] $utterance_1$ [SEP] $utterance_2$ [SEP] , ... , SEP]"*. The training objective function is the log-likelihood of the correct span.

4 Experiments

4.1 Experiments on Mention Detection

For mention detection, we evaluate our model on datasets of the anaphora resolution in dialogue shared-task. In dev-phase, For example, when we evaluate *light* dataset, we train the rest of datasets except *light*. Same manner is used for evaluating other datasets. In test-phase, we train all of the dev datasets. We use precision, recall and F1 socres for

Datasets	Precision	Recall	F1-score
light	94.09	91.25	92.67
AMI	69.29	54.04	85.55
Persuasion	93.89	91.19	92.54
Switchboard	89.46	86.76	88.11
ARRAU	84.98	93.74	89.36

Table 2: Results on mention detection for dev datasets.

Datasets	Precision	Recall	F1-score
light	95.36	93.00	94.18
AMI	92.12	89.64	90.88
Persuasion	92.91	92.95	92.93
Switchboard	92.70	89.43	91.07

Table 3: Results on mention detection for test datasets.

mention detection evaluation. As shown in Table 3, our mention detection model shows 94.18% F1 scores in the light dataset. 90.88 in the AMI dataset, 92.93 in the Persuasion dataset, and 91.07 in the Switchboard dataset.

4.2 Experiments on Coreference Resolution

		Light	AMI	Persuasion	Switchboard	ARRAU
	P	87.6	70.7	80.8	80.7	81.6
MUC	R	70.8	44.6	72.0	67.4	65.9
	F1	78.2	54.7	76.1	73.4	73.0
	P	65.0	63.7	66.1	64.5	67.5
Bcub	R	55.0	43.3	67.7	60.6	53.2
	F1	59.6	51.6	66.9	62.5	59.5
	P	53.0	41.9	61.7	54.0	36.3
CEAFe	R	78.1	70.1	75.4	75.5	75.9
	F1	63.1	52.4	67.9	62.9	49.1
CoNLL		67.0	52.9	70.3	66.31	60.5

Table 4: Results on coreference resolution for dev datasets.

		Light	AMI	Persuasion	Switchboard
	P	89.3	69.0	76.5	80.9
MUC	R	76.1	53.5	78.4	66.7
	F1	82.1	60.3	77.3	73.1
	P	64.9	63.6	65.6	66.2
Bcub	R	56.1	49.8	72.6	58.7
	F1	60.2	55.9	68.9	62.2
	P	55.7	46.5	61.9	52.0
CEAFe	R	78.4	72.4	73.1	75.8
	F1	65.1	56.6	67.0	61.7
CoNLL		69.1	57.59	71.0	65.67

Table 5: Results on coreference resolution for test datasets.

Table 4 and Table 5 shows the results of the scores of MUC Vilain et al. (1995), B-cubed Bagga and Baldwin (1998), CEAF Luo (2005), and the averaged CoNLL score Pradhan et al. (2014). As shown in Table 5, our model shows 69.1% in the light, 57.5% in the AMI, 71.0% in the Persuasion and 65.6% in the Switchboard respectively. Our model achieved the overall average score 65.9% (ranked at top 3 in subtask1) of all datasets.

4.3 Experiments on Bridging Reference

In test-phase of bridging reference, there are two phases depending on whether the gold mention is given. When the gold mention is given in Eval-Br(Gold) phase, but in Eval-Br(Pred), we need to extract the mentions. In Eval-Br(Pred) phase, we use our mention detection model and coreference resolution model. In Eval-Br(Gold) phase, we only use our coference resolution model. In other words, we use the result of in the subtask1. For evaluating bridging reference, we use precision, recall and f1-score of AR, FBM and FBE. AR is anaphora recognation, FBM is full bridging at mention level and FBE is full bridging at entity level. Table 6 shows the results of the Eval-Br(Pred) phase. In FBE scores, recall is significantly higher than precision. One possible reason is that our bridging reference MRC model can not classfy no-answerable query. Even if a no-answerable query is given to our model, it is inevitable to predict answer span since our model does not train to classify whether a query is answerable. Table 7 shows the results of the Eval-Br(Gold) phase. In FBE scores, precision is significantly improved compare with the Eval-Br(Pred) phase. It suggests that the correct mention detection is effective for coreference resolution and bridging reference. Our model achieved the overall average score 11.76% (ranked at top 2) and 17.27% (ranked at top 2) in Eval-Br(Pred) and Eval-Br(Gold) respectively.

		Light	AMI	Persuasion	Switchboard
	P	28.7	32.3	20.9	29.7
AR	R	54.6	38.3	60.4	43.4
	F1	37.7	35.0	31.0	35.2
	P	7.2	7.1	6.5	6.8
FBM	R	14.9	9.1	20.1	10.7
	F1	9.7	8.0	9.8	8.3
	P	10.2	9.4	8.2	9.2
FBE	R	19.5	11.2	23.9	13.5
	F1	13.4	10.2	12.3	10.9

Table 6: Results on bridging reference in Eval-Br(Pred) phase.

		Light	AMI	Persuasion	Switchboard
AR	P	38.1	41.9	31.2	41.1
	R	34.6	30.8	53.1	30.9
	F1	36.3	35.5	39.3	35.3
FBM	P	11.9	12.9	11.7	14.9
	R	12.2	10.9	22.5	13.1
	F1	12.0	11.8	15.4	14.0
FBE	P	17.5	18.0	14.9	21.3
	R	15.9	13.2	25.3	16.0
	F1	16.6	15.3	18.7	18.3

Table 7: Results on bridging reference in Eval-Br(Gold) phase.

5 Conclusion

We proposed the pipeline model for resolution of anaphoric reference and resolution of entity reference. Our resolution of anaphoric reference model consists of mention detection model and coreference resolution model. Mention detection model extracts mentions via the parentheses prediction. Based on the result of mention detection model, the mention representation is created and then is fed to the coreference resolution model. Our coreference resolution model points to the closest mention using pointer network, and then merges mentions in same cluster sequentially. In subtask1, Our model achieved 65.9% the overall average score of all datasets (ranked at top 3). Our resolution of entity reference model utilizes a MRC framework. Given a query for related entity, our model predicts the related entity span. Our model achieved the overall average score 11.76% (ranked at top 2) and 17.26% (ranked at top 2) in Eval-Br(Pred) and Eval-Br(Gold) respectively.

Acknowledgements

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques). Also, This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1F1A1069737).

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*, volume 1, pages 563–566. Citeseer.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4115–4128.

Pradeep Dasigi, Nelson F Liu, Ana Marasović, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. *arXiv preprint arXiv:1908.05803*.

Sopan Khosla, Yu Juntao, Manuvinakurike Ramesh, Ng Vincent, Poesio Massimo, Strube Michael, and Rosé Carolyn. 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*.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. *arXiv preprint arXiv:1905.05529*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2014, page 30. NIH Public Access.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134*.