Gender Balanced Coreference Resolution

Nicholas Tan
Stanford University
Stanford, CA 94305
ntan2012@stanford.edu

Hongshen Zhao
Stanford University
Stanford, CA 94305
zhhongsh@stanford.edu

Abstract

There has been significant progress in the NLP effort of coreference resolution. In particular, Kenton Lee et. al have developed a fully end-to-end model that achieves good performance on LDC's Ontonotes dataset. However, there has been recent a recent interest in identifying and designing to address bias in machine learning models. In particular, these studies have exposed gender imbalances in the coreference resolution task due to the gender imbalance in existing corpora and datasets. Our contribution has been to integrate Google's recently released gender balanced dataset into Kenton Lee's fully deep end-to-end model for coreference resolution.

1 Introduction

1.1 Task

Co-reference resolution is the task of identifying all words and phrases that refer to the same entity in a corpus of text. In particular, to accomplish this task, one must find all pronouns or referring expressions and connect them to their antecedents. In natural language, these connections may be ambiguous and resolving them may rely heavily on contextual clues or implied past experiences. This makes the task especially difficult in machine understanding of natural language, where human-level performance has not yet been achieved.

For example, given the following source text and ambiguous pronoun (**bolded**):

> Kathleen Nott was born in Camberwell, London. Her father, Philip, was a lithographic printer, and her mother, Ellen, ran a boarding house in Brixton; Kathleen was their third daughter. **She** was educated at Mary Datchelor Girls' School (now closed), London, before attending King's College, London.

Our objective is to find the antecedent for **She**, which is **Kathleen**. Note that it may be ambiguous in certain interpretations of the text whether **Ellen** may have been the true antecedent.

2 Related Work

Kenton Lee's et. al. seminal paper introduces the first complete co-reference resolution model trained end-to-end. The task of co-reference resolution is to identify antecedants to potentially ambiguous referring words or phrases (e.g. pronounts). The innovation being accomplished in this co-reference resolution work by Kenton Lee et. al. is using a completely end-to-end model for learning. Explicitly, they have built a model that utilizes and integrates deep learning at every step. Historically, co-reference resolution was accomplished with a large preprocessing step to parse each sentence for analysis.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

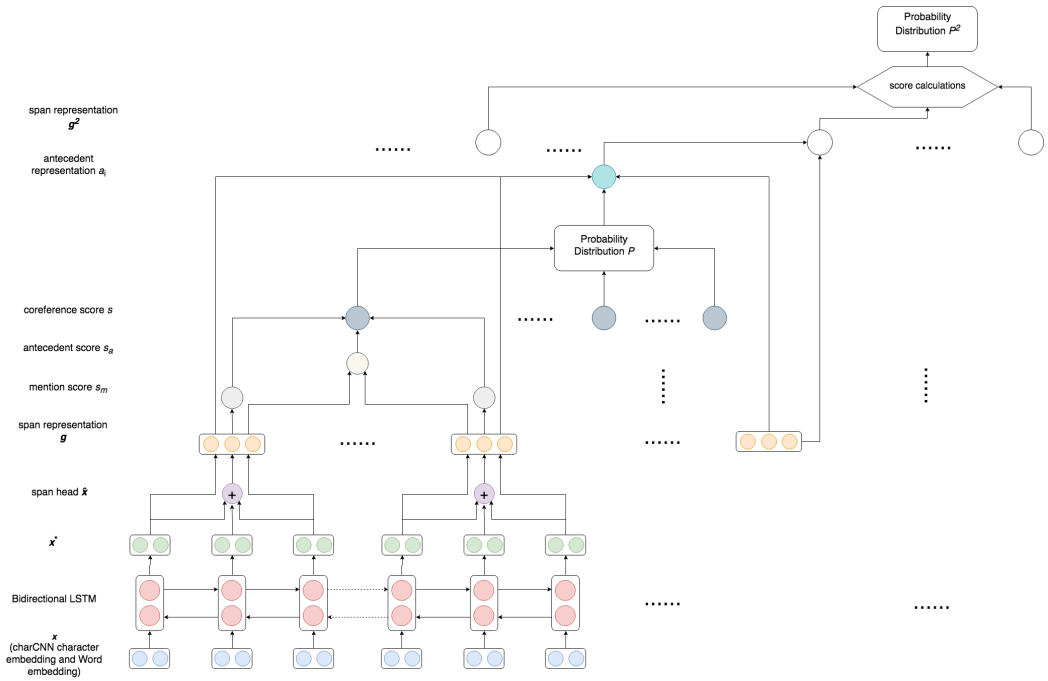

Figure 1: End-to-end coreference architecture with 2nd order coreferenec resolution - computes embedding representations of spans to predict more likely spans for antecedent matches

Kenton Lee et. al. have showed that performance could be significantly improved without using syntactic parsers. Their results outperformed existing models by 1.5 F1 on the OntoNote benchmark and by 3.1 F1 using a 5-model ensemble.

In (4), Wiseman illustrated an example of consistency errors where the predicted clusters are locally consistent but globally inconsistent. To overcome this problem, Lee et. al. proposed a higher-order coreference resolution improvement together with a coarse-to-fine antecedent pruning in (2). The coarse-to-fine pruning reduces the additional computation cost from multiple interactions in the higher-order resolution. The higher-order resolution helps to model to condition on higher-order structures in the documentation. In this project, we use a 2nd-order coreference resolution which is shown to have the best performance without incurring too much additional overhead in computation.

3 Approach

3.1 Architecture

We leverage work by Kenton Lee et. al. in coreference resolution using end-to-end models (2), (1). They addressed the co-reference resolution task as the set of decisions to assign an antecedent y_i for every possible span i in the document, where $y_i \in Y(i) = \{\epsilon, 1, ..., i-1\}$. The assignment decisions can be made by consulting a distribution $P(y_i)$ over antecedents for each span i, which is learnt by the model. ϵ is a dummy antecedent representing two possible scenarios: 1). the span is not an entity mention or 2). the span is an entity mention but it is not coreferent with any previous span. These assignment decisions implicitly define a final clustering, which can be recovered by grouping all spans that are connected by a set of antecedent predictions. Their solution leveraged several neural network models - they used a bidirectional LSTM and a CNN for word representations and information encoding. A CharCNN is used to perform character level embeddings and to recover out-of-vocabulary words. To replace the use of syntactic parsers, they used an attention mechanism over words in each each span to learn a task-specific notion of headedness. Because computing over all possible spans in the text would be expensive, a pruning mechanism was also built to identify more probable spans and discard irrelevant ones (See Figure 1).

Embeddings: We use GloVe and Turing embeddings to construct vector representations for each word. A CharCNN is also used to account for encountered words that are not in the vocabulary.

Span Representations (LSTM): To compute vector representations of each span, bidirectional LSTMs are used to encode every word in the context. The LSTM equations are as follows:

$$\mathbf{f}_{t,\delta} = \sigma(\mathbf{W}_f[\mathbf{x}_t, \mathbf{h}_{t+\delta,\delta}] + \mathbf{b}_i)$$
$$\mathbf{o}_{t,\delta} = \sigma(\mathbf{W}_o[\mathbf{x}_t, \mathbf{h}_{t+\delta,\delta}] + \mathbf{b}_o)$$
$$\widetilde{\mathbf{c}}_{t,\delta} = \tanh(\mathbf{W}_c[\mathbf{x}_t, \mathbf{h}_{t+\delta,\delta}] + \mathbf{b}_c)$$
$$\mathbf{c}_{t,\delta} = \mathbf{f}_{t,\delta} \circ \widetilde{\mathbf{c}}_{t,\delta} + (1 - \mathbf{f}_{t,\delta}) \circ \mathbf{c}_{t+\delta,\delta}$$
$$\mathbf{h}_{t,\delta} = \mathbf{o}_{t,\delta} \circ \tanh(\mathbf{c}_{t,\delta})$$
$$\mathbf{x}^* = [\mathbf{h}_{t,1}, \mathbf{h}_{t,-1}]$$

Deep contextualized word representations (3) can be used at the input to the LSTMs to improve performance.

Attention mechanism: An attention mechanism to learn a task-specific notion of headedness over words in each span is used to replace syntactic parsers which would directly output the head of a span

$$\alpha_t = \omega_\alpha \cdot \text{FFNN}_\alpha(\mathbf{x}_t^*)$$
$$a_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp \alpha_k}$$
$$\hat{\mathbf{x}}_i = \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot \mathbf{x}_t$$

FFNN denotes a feed-forward neural network that computes a nonlinear mapping from input to output vectors. The above span information is concatenated with a feature vector $\phi(i)$ to produce the final representation $\mathbf{g}_i$ of span i:

$$\mathbf{g}_i = [\mathbf{x}^*_{\text{START}(i)}, \mathbf{x}^*_{\text{END}(i)}, \hat{\mathbf{x}_{(i)}}, \phi(i)]$$

Scoring architecture: a score is computed using the vector representations for each relevant span with feed-forward networks

$$s_m(i) = w_m \cdot \text{FFNN}_m(\mathbf{g}_i)$$
$$s_a(i,j) = w_a \cdot \text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i,j)])$$

The antecedent scoring function $s_a(i,j)$ consists of the element-wise similarity of $\mathbf{g}_i$ and $\mathbf{g}_j$ calculate by element-wise multiplication, and a feature vector $\phi(i,j)$ encoding speaker and genre information from the metadata and the distance between the two spans. The mention scores are used to prune the space of spans and antecedents, only top M spans is considered for coreference decisions based on the mention score $s_m(i)$.

The outcome of the learning is a ditribution $P(y_i)$ over antecedents for each span i:

$$P(y_i) = \frac{e^{s(i,y_i)}}{\sum_{y \in Y(i)} e^{s(i,y)}}$$

Coarse-to-fine antecedent pruning: This pruning is used to further reduce the computational cost in higher order coreference resolutions. We use a bilinear score

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

to prune antecedents with low likelihood. This is achieved by keeping top K antecedents based on the score $s_m(i) + s_m(j) + s_c(i,j)$ of the remaining M spans. Instead of simply replacing the scoring function with s_c, we can just include s_c as an additional factor in the score. This helps to adjust the likelihood of the antecedent to the span.

$$s(i,j) = s_m(i) + s_m(j) + s_c(i,j) + s_a(i,j)$$

Second-order Coreference Resolution: Introduced in (2), higher-order resolutions can be used to solve consistency errors demonstrated by the example in (4) where The plurality of **[you]** is underspecified, making it locally compatible with both **[I]** and **[all of you]**, while the full cluster would have mixed plurality, resulting in global inconsistency. As shown in (2), the second-order coreference gives the best performance. We first calculate the expected antecedent epresentation for the current span i using the current antecedent distribution $P^1(y_i)$.

$$P^1(y_i) = \frac{e^{s(\mathbf{g}_i, \mathbf{g}_{y_i})}}{\sum_{y \in Y(i)} e^{s(\mathbf{g}_i, \mathbf{g}_y)}}$$

$$\mathbf{a}_i = \sum P^1(y_i) \cdot \mathbf{g}_{\mathbf{y_i}}, \mathbf{f}_i = \sigma(\mathbf{W}_f[\mathbf{g}_i, \mathbf{a}_i])$$

The span representation $\mathbf{g}_i$ for the span i is then updated via interpolation with its expected antecedent representation $\mathbf{a}_i$

$$\mathbf{g}_i^2 = \mathbf{f}_i \circ \mathbf{g}_i + (\mathbf{1} - \mathbf{f}_i) \circ \mathbf{a}_i$$

The refined antecedent distribution is calculated by

$$P(y_i) = P^2(y_i) = \frac{e^{s(\mathbf{g}_i^2, \mathbf{g}_{y_i}^2)}}{\sum_{y \in Y(i)} e^{s(\mathbf{g}_i^2, \mathbf{g}_y^2)}}$$

The original model is used to initialize the span representation $\mathbf{g}_i$. We first calculate the expected antecedent $\mathbf{a}_i$ using the current antecedent distribution $P^1 y_i$. A gate vector $\mathbf{f}_i$ is then used to determine for each dimension whether to keep the current span information or to update it with the information from the expected antecedent $\mathbf{a}_i$ using interpolation.

2-stage pruning: The candidate spans and antecedents will go through a 2-stage pruning before the final 2nd-order coreference resolution computation. In the first pruning stage, only the top M spans will be kept based on the mention score $s_m(i)$. In the second pruning stage, only the top K antecedents of each remaining span i from the first pruning stage will be kept based on the score $s_m(i) + s_m(j) + s_c(i,j)$.

3.2 Baseline Implementation

For baseline evaluation of gender imbalance, we sourced the standard (gender-unbalanced) dataset from LDC - CoNLL 2012 (http://conll.cemantix.org/2012/data.html).

The baseline model architecture is the state-of-art end-to-end coreference annotator by Kenton Lee et. al. The model description is detailed in the above section and in their paper (http://aclweb.org/anthology/D17-1018). This model was reproduced, trained, and evaluated on a gender-imbalanced dataset. We have cloned and worked beginning from Kenton Lee's implementation (https://github.com/kentonl/e2e-coref) as referenced in his paper (2). We adapted the implementation to work with tensorflow 1.9. On top of the baseline model we experimented with different sub-network architectures including our addition of a residual skip-layer feed-forward network.

We have used the same model as a skeleton to augment and train on a different gender-balanced dataset.

3.3 Dataset

Our baseline evaluation used the English coreference resolution annotations from the CoNLL-2012 shared task benchmark dataset. This dataset contains 2802 training documentations, 343 development documentations, and 348 test documentations. The training documentations contain on average 454 words and a maximum of 4009 words.

For our targeted study, we used the Google GAP Conference dataset, which has 8,908 coreference-labeled pairs of (ambiguous pronoun, antecedent name). The dataset has been specifically designed

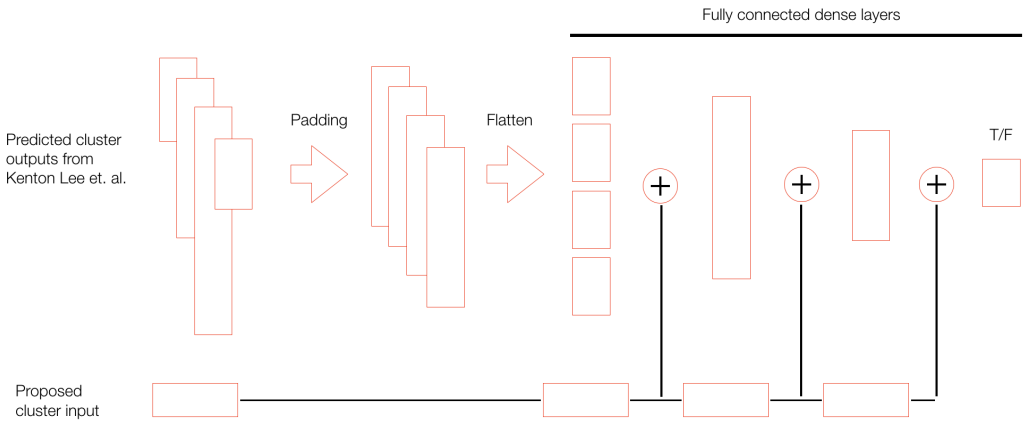

Figure 2: Residual dense skip-connection layers showing the integration of coreference clusters as input to the model

to represent the challenges of resolving ambiguous pronouns in a gender-balanced context. The dataset samples each excerpt from various Wikipedia articles and annotations are created with human labeling. The dataset is provided by the Google AI Language group (https://github.com/google-research-datasets/gap-coreference).

There are various differences between the CoNLL-2012 shared task dataset and the Google GAP conference dataset. For the coreference task, the CoNLL-2012 dataset provides, as input, text paragraphs for training and, as output, span clusters for coreference annotation as labels. The Google GAP dataset however provides, as input, text paragraphs accompanied by a proposed span cluster for training and, as output, a label of TRUE or FALSE indicating whether that cluster is a valid coreference pairing.

3.4 Integration of GAP dataset

Because the original CoNLL-2012 dataset used to train the model differed from the gender balanced dataset in structure, the model architecture needed updates to accommodate the new input-output labeling scheme. In particular, proposed cluster spans were now accepted as inputs into the model instead of predicted as outputs. A TRUE or FALSE label would be the new output.

In the spirit of maintaining the end-to-end nature of Kenton Lee's model, several fully connected layers were appended to the predicted span outputs with deep residual skip-connections to compare the proposed span cluster with the original output from Kenton Lee et. al. These arbitrary length output sequences were padded up to a constant size before entering the fully connected layers; and after each fully connected layer, dropout was performed. Each of the layers used a ReLU activation, except for the final layer, which used a sigmoid to output a probability between 0 and 1. The layer sizes were chosen to continually decrease until one node was reached, which would be used to predict the TRUE or FALSE labeling accompanying each example in the Google GAP dataset. The loss function was thus edited to be the binary cross entropy loss instead of the previous softmax loss.

4 Experiments

4.1 Evaluation

Accuracy was assessed by the F1 score, which is the harmonic mean of precision and recall. Explicitly, the score is calculated as

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

	Average F1
Clark and Manning	55.0
Kenton Lee et. al. (2013)	50.5
Kenton Lee et. al. (2017)	64.7
Our updated implementation for GAP	58.2

Table 1: Average F1 scores

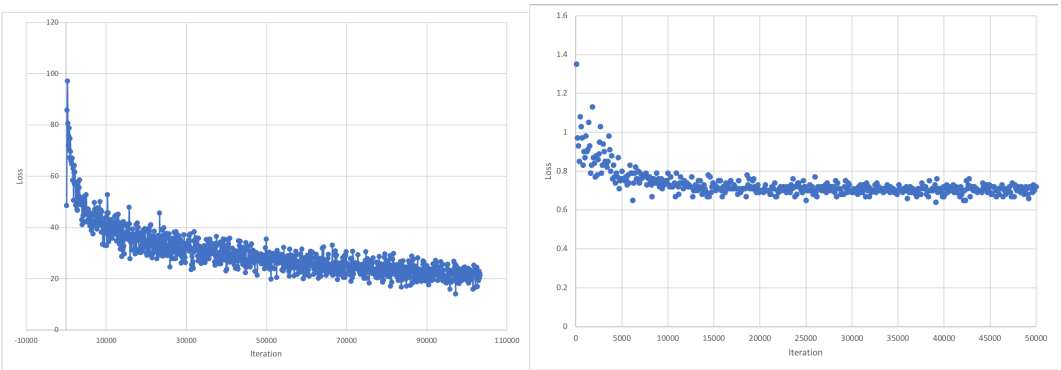

Figure 3: Plot of loss showing model convergence during training. Left - original model using softmax loss, Right - model updated for Google GAP dataset using binary cross entropy loss

where precision is the ratio of true positive guesses to total positive guesses and recall is the ratio of true positive guesses to total ground truth positives. We compute the F1 scores for the standard MUC, B^3, $CEAF_{\phi 4}$ metrics. The main evaluation metric is the average F1 of the three metrics.

4.2 Parameter Tuning

Network configurations: The bidirectional LSTMs have three layers and each hidden state has 300 dimensions. The feed-forward network has 2 hidden layers and 150 dimensions. In the CharCNN network, the characters are represented as learned 8-dimensional embeddings. The convolutions have kernel sizes of 3, 4, and 5. Each 1D-Conv layer consists of 50 filters. The dropout rate for the LSTM outputs is 0.4. The dropout for the word embeddings and CharCNN outputs is 0.5. A dropout rate of 0.2 is applied to all the other hidden layers.

The three layer sizes in our skip-connection dense funnel were 1000, 100, and 1. The dropout probabilities after each hidden layer were 0.5 and 0.1.

Gradient Clipping: To avoid gradient explosion, we apply global clippings based on gradient norm of max value 5.

Prunning: We used a max span width of $L = 30$, the number of spans per word $\lambda = 0.4$ and the maximum number of antecedents $K = 50$ to prune the spans.

Optimizer: The optimizer used is Adam optimizer. The learning rate is 0.001 with a decay rate of 0.999 and a decay frequency of 100 steps.

4.3 Results

Our implementation of the original model achieved an F1 score of 0.7283 on its original dataset. The average precision was 75.76% and the average recall was 70.11%.

Our implementation of the updated model achieved an F1 score of 0.5818 on the Google's GAP dataset. The average precision was 58.19% and the average recall was 56.63%.

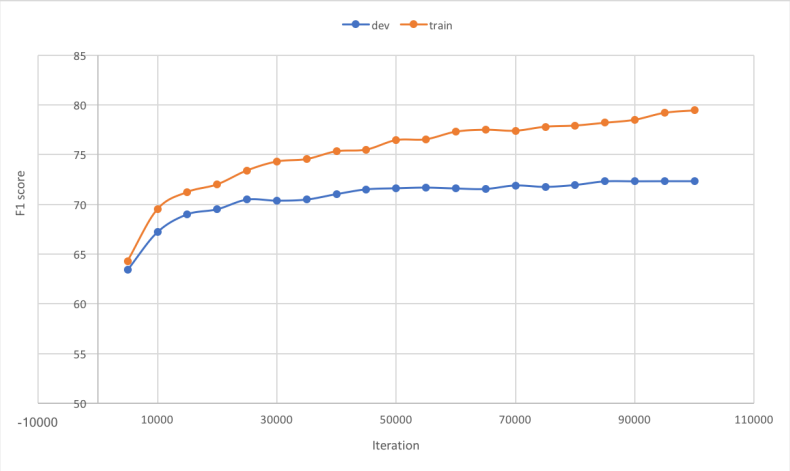

Figure 4: Plot of the F1 performance metric at each epoch, computed on the dev set and the training set, showing model learning during training

5 Analysis

As shown in Table Table 1, our model trained on the GAP dataset fails to improve the average F1 score compared to the off-the-shelf end-to-end coreference resolver with coarse to fine coreference resolution on the GAP development set. The primary reason for this is because the original OntoNotes training data has richer information in each training example. There are significantly more clusters and each cluster has richer coreference structures, giving multiple spans that refer to the same object. However, in the GAP data set, only one pronoun and one antecedent are given for each training example.

Incorrect coreference resolution example: When onlookers expressed doubt, claiming that the Proctor family was well regarded in the community, the girl promptly came out of her trance and told them it was all for "sport". On March 29, 1692, Abigail Williams and Mercy Lewis again said they were being tormented by Elizabeth's spectre. A few days later, Abigail complained that Elizabeth was pinching her and tearing at her bowels, and said she saw Elizabeth's spectre as well as John's.

Correct coreference: The correct antecedent of 'her' is 'Abigail'.

Possible reason: The model is restricted to lowered order structure resolutions where it tries associating determiners to the words with the form of xx's. Although the gender was not resolved correctly, 'her' not being coreferent to 'John's', the limited information in the GAP data set does not help the model to learn more complex relationships between nouns.

Incorrect coreference resolution example: Meet Mike, the shortest bully to appear on the show. He stands at 5'4 and is a Bronx gym rat who makes life miserable for his victims, Lorenzo and Joey. Mayhem Miller brings in MMA fighter Eddie Alvarez with a record of 22 wins and 2 losses to teach him a lesson.

Correct coreference: 'him' is not coreferent to anything in the text according to the GAP labeling.

Possible reason: The model is poor at dealing with coreference resolution with ambiguities. The coreference cluster ['him', 'MMA fighter Eddie Alvarez'] is correct grammatically but incorrect when we consider all the context information in the text. I would say the model did it best for this inference but to predict correctly for this example, it needs to learn far more than linguistic rules including logic analysis ability.

6 Conclusion

Given the prevalence of NLP systems in modern-day automated decision making and user interaction, it has become paramount to audit deep neural networks for inherent bias. In particular, datasets

used to train our models have been shown to possess gender imbalances that cause imbalanced performance. Our contributions in this project were to integrate a gender-balanced dataset into a state-of-the-art end-to-end coreference resolution system.

Limited by the information available in GAP data set itself, the model trained with GAP did not perform as well as the model trained with the original OntoNotes data set. This drastically impairs the ability of our model to capture various forms and structures in coreference resolution, like the first example shown in Section 5. Another challenging factor in the GAP data set is its ambiguity. Like the second analysis example in Section 5, many texts are ambiguous that you can find coreferences grammatically correct but incorrect given the context information. To predict correctly, the model needs to have the ability to understand the text meaning which exceeds the scope of the problem we are trying to solve.

7 Additional Information

Mentor: Xiaoxue Zang

References

[1] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. End-to-end Neural Coreference Resolution. *arXiv e-prints*, page arXiv:1707.07045, Jul 2017.

[2] K. Lee, L. He, and L. Zettlemoyer. Higher-order Coreference Resolution with Coarse-to-fine Inference. *arXiv e-prints*, page arXiv:1804.05392, Apr 2018.

[3] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv e-prints*, page arXiv:1802.05365, Feb 2018.

[4] S. Wiseman, A. M. Rush, and S. M. Shieber. Learning global features for coreference resolution. *CoRR*, abs/1604.03035, 2016.