Fine-Tuning XLNet for Coreference Resolution

Poonam Parhar, Delon Saks
{poonamparhar, delonsaks}@berkeley.com

Abstract

Coreference Resolution is the task of clustering all expressions that relate to the same entity in a text, and is highly important for various high level NLP tasks dependent on natural language understanding such as question answering and document summarization. In this work, we apply XLNet (Zhilin et al., 2019) to the Coreference Resolution task. XLNet is an autoregressive pre-training method that has the capability of learning bidirectional contexts, and has shown to improve performance significantly for tasks involving longer text sequences. This work explores various ways to achieve better performance on the coreference resolution task by incorporating larger contexts into the model. For this task, our XLNet-Base baseline model achieved an average F1 score of 65.23%, and after incorporating longer contexts we were able to improve the performance remarkably and achieve an F1 score of **72.47%**.

1. Introduction

Recently, XLNet (Zhilin et al., 2019) - a generalized autoregressive pretraining model - reported that it outperforms BERT on 20 tasks, often by a large margin, including question answering, natural language inference, sentiment analysis, and document ranking. However, it did not report any results for coreference resolution. One of the notable strengths of XLNet is that it does not have any limitation on the size of input sequences, and can accept token sequences longer than 512. This motivated us to experiment and determine its performance for the coreference resolution task, hoping that using longer contexts would help improve the performance of this task.

In this paper, we fine-tune XLNet to coreference resolution on the OntoNotes corpus (Pradhan et al.,2012), described going forward as **coref-xlnet**. The state of the art for coreference resolution, **coref** model of SpanBert in Mandar et al.(2020) extended the **c2f-coref** model in Lee et al.(2018) using two methods - independent and overlap. The *independent* variant of coref uses non-overlapping segments, and each segment is fed as an independent instance to SpanBert. The *overlap* variant splits the document into overlapping segments so as to provide the model with context beyond 512 tokens. In this work, we extend the *independent* variant of the **coref** model to fine-tune it with XLNet, and conduct various experiments to improve the performance by providing longer contexts to our model. We start with our baseline model; fine-tuned with XLNet-Base using a sequence length of 128 and a batch size of 3, which resulted in an average F1 score of 65.23% for this task. By providing longer contexts to our model, we were able to improve its performance significantly, achieving an average F1 score of **72.47%**.

2. Background and Method

For our experiments, we extend the higher-order coreference model, **coref** of Mandar et al.(2020), which is the current state of the art for the English OntoNotes dataset (Pradhan et al.,2012), reporting an average F1 score of 79.6% with SpanBert-Large, and 77.7% with the SpanBert-Base model on the coreference resolution task. Prior to coref, **c2f-coref** in Lee et al.(2018) which was an extension of **e2e-coref** in Lee et al.(2017) had presented the state of the art in coreference resolution.

Background: Coreference resolution is the task of clustering mentions in a text which refer to the same real-world entities. We evaluate document-level coreference resolution on the CoNLL-2012 shared task (Pradhan et al.,2012). We use the *independent* variant of the Joshi et al.(2019b) implementation of the higher-order coreference model

(Lee et al., 2018). In the independent method, a document is divided into *non-overlapping* segments of a predefined length. Each segment is then encoded independently by the pre-trained XLNet transformer which replaces the original SpanBert encoder. Next, the XLNet embeddings are provided to the upper coref-xlnet layers, and the entire network is fine-tuned on the OntoNotes corpus.

Applying XLNet: We replace the SpanBert encoder in the coref model with the XLNet transformer. Documents are split into segments of *max_segment_len* chosen from [128, 256, 384, 512, 640, 768], which is treated as a hyperparameter. We use the *Independent* variant for splitting the documents, which uses non-overlapping segments of documents, each of the segments acts as an independent instance for XLNet. We treat the first and the last word representation of a segment concatenated with the attended version of all words in the span as span representations.

3. OntoNotes Corpus: Document Level

We use **OntoNotes Release 5.0** corpus for fine-tuning XLNet to perform coreference resolution. It provides a large corpus with several layers of accurate and integrated annotation, with one of the layers proving a corpus for general anaphoric coreference. We pre-process and split the data into training, dev and test partitions using the scripts made available by the CoNLL-2012 shared task. The primary evaluation metric is the average F1 of three metrics - MUC, B3, and CEAFE on the test set computed by the official CoNLL-2012 evaluation scripts.

4. Models

XLNet is a generalized autoregressive (AR) pretraining method that learns from bi-directional context of words sequences. It employs the **Permutation Language Modeling** objective for pre-training that has shown great results. It leverages the best of both AR language modeling and AE while avoiding their limitations. Our work aims to explore various ways exploiting the strengths of XLNet to improve the performance of the coreference resolution task. Figure 1 shows the architectural design of **coref** and our model **coref-xlnet**:

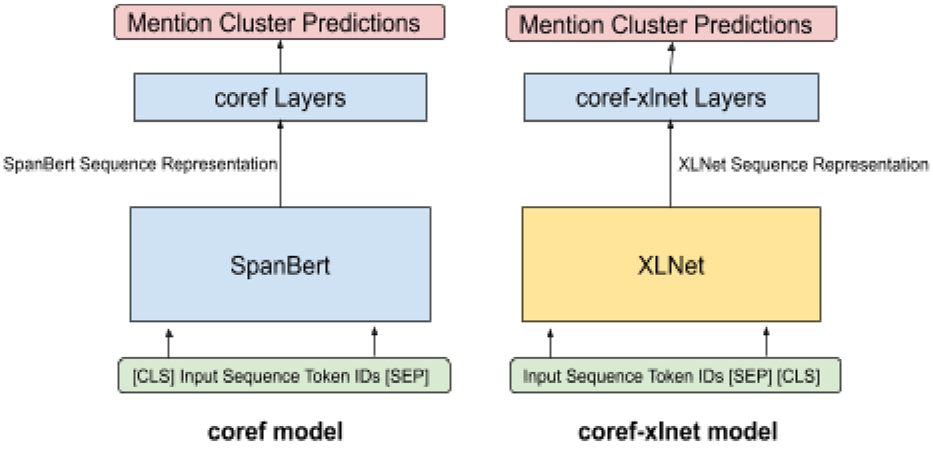

Mention Cluster Predictions
coref Layers
SpanBert Sequence Representation
SpanBert
[CLS] Input Sequence Token IDs [SEP]
coref model
Mention Cluster Predictions
coref-xlnet Layers
XLNet Sequence Representation
XLNet
Input Sequence Token IDs [SEP] [CLS]
coref-xlnet model

Figure 1: Architectural design of coref and coref-xlnet

5. Exploratory Data Analysis

We pre-process and split the OntoNotes corpus into training, dev and test set using the CoNLL-2012 shared task scripts that create *_gold_conll files for the English language. Next, we process these gold files to create examples in json format for the XLNet model. There are 8406, 1029 and 1044 documents in the training, dev and test sets respectively.

We examine the examples tokenized using XLNet's **SentencePiece** tokenizer to ensure that the segments had '[SEP]' and '[CLS]' appended at the end for the XLNet pretrained model. From the tokenized examples, mention clusters are created to be used as labels for the supervised learning of our XLNet based coref-xlnet model.

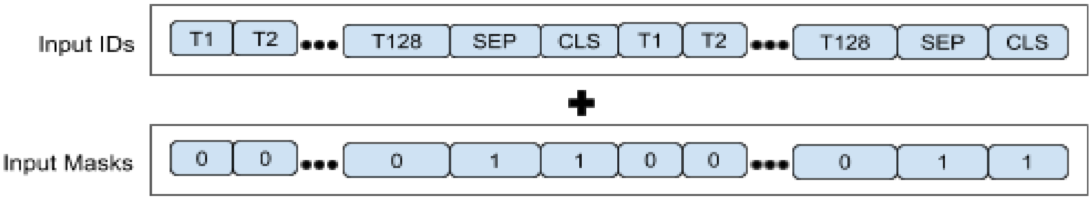


Figure 2: An input example for XLNet of batch size 2, and each batch having 128 tokens. Each input sequence is separated by SEP and CLS tokens. Corresponding input masks are also supplied to XLNet model, 0 indicating a valid token and 1 representing a special token or padding.

6. Experimental Setup and Hyperparameters

We extend the *Independent* variant of the coref model, and replace the underlying SpanBert method with the XLNet. For tokenization, we use the XLNet's SentencePiece model. In our experiments, we use XLNet-Base to produce sequence embeddings and then prepare span representations from those embeddings. We trained several XLNet-Base based coref-xlnet models using a range of values for *max_segment_len*, and used different learning rates for these sequence lengths. We had access to a Google-Colab machine with **Tesla V100-SXM2-16GB GPU**. We also attempted to use TPUs for fine-tuning our models so as to have larger memory resources, but could not successfully use them.

For all our models, we used *train_steps* of 80K, *warmup_steps* of 500, and a dropout rate of 0.2. For *max_segment_len* of 128 and 256, we used the *learning_rate* of 2e-05. For *max_segment_len* of 384 a learning_rate of 3e-05, and for segments having length 512 or greater, a learning_rate of 5e-05 gave us the best observed performance. We attempted to use the *weight_decay* hyperparameter in order to perform weight decay for parameters with AdamOptimizer, but due to the limited memory resources we encountered out-of-memory errors, and could not confirm if using it would have improved the performance or not.

7. Baseline Model: coref-xlnet

The coref-xlnet model fine-tuned with **XLNet-Base** trained using the maximum segment length of 128 and a batch size of 3 is our baseline model. SpanBert-Base based coref also used a batch size of 3 for fine tuning. In Table 1 below, we provide the performance results of our baseline model.

Segment Length	Average F1 %	Precision %	Recall %
128	65.23%	73.04%	58.97%

Table 1: Baseline model trained with max_segment_len of 128 and a batch size of 3.

In Table 2 below, for reference, we present the evaluation results of SpanBert-Base based coref model:

Segment Length	Average F1 (%)	Precision	Recall
128	74.90%	77.08%	72.84%
256	76.64%	78.26%	75.09%
384	**77.71%**	**78.70%**	**76.75%**
512	77.56%	78.51%	76.62%

Table 2: Performance of **coref** fine-tuned with **SpanBert-Base** using different max_segment_len values.

8. Improvement Experiments: Larger Context

We conducted various experiments to improve the performance of our baseline model. In order to provide larger context to models, we first ran experiments by increasing the length of the text sequences fed into models. Next, we experimented with increasing the number of sequences, the batch size, used for fine-tuning the models.

Increasing maximum segment length: We fine-tuned models with various max_segment_len values using a batch size of 3, and observed that the performance increased as we increased the sequence length. Note that for the experiments with segment lengths greater than 512, we had to decrease the batch size due to the availability of limited accelerator memory resources. We observe (Table 3) that the performance of our models improves significantly as we make larger contexts (Segment Length x Batch Size) available to XLNet.

Segment Length	Batch Size	Average F1 %	Precision %	Recall %
128	3	65.23%	73.04%	58.97%
256	3	68.25%	76.70%	61.51%
384	3	70.08%	**77.62%**	63.90%
512	3	70.43%	75.67%	65.89%
640	2	70.16%	75.80%	65.32%
768	2	**72.47%**	72.71%	**72.23%**

Table 3: Performance with various max_segment_len values evaluated against **coref-xlnet** fine-tuned with **XLNet-Base**.

Increasing batch sizes: We also conducted experiments with various batch sizes (controlled with *max_training_sentences* hyperparameter). Due to the high memory requirements with larger batch sizes, and availability of limited memory resources on our training machine, we could use a batch size of upto 11 sequences with 128 segment length, and a batch size of only 6 with 256 segment length. We could not use more than 3 sequences for our models with 384 and 512 segment length, and could only use a batch size of 2 with segment lengths 640 and 768.

Segment Length	Batch Size	Average F1 %	Precision %	Recall %
128	11	66.84%	71.98%	62.42%
256	6	68.60%	76.65%	62.10%

Table 4: Performance results with increased batch sizes for segment lengths 128 and 256

These results indicate that providing a larger batch of training sequences to XLNet also helps to improve the overall performance of models. Given the above observations, we conjecture that if we could increase the batch size for our models having longer token sequences, they would yield better results.

9. Error Analysis

Weaknesses: With our best model, we performed a qualitative analysis on the OntoNotes English test set results. First, we focussed on the recall score, which means on the mention-clusters that are left out and are not predicted by our model. The analysis suggests that most of the missed predictions involve cases requiring mention paraphrasing and pronouns resolution. Table 5 shows examples of these.

[('monopoly profits of the monopoly companies'), ('This part of money')]	Fails to match paraphrased mentions
[('the former'), ('the question you make another deal are you better off with the deal')]	
[('the Radio Canada International test transmissions to your part of the world'), ('They')]	Fails to match phrases with their corresponding pronouns
[('them'), ('them'), ('they'), ('these people'), (‘'these people out there perpetrating these crimes')]	

Table 5: Example clusters that are missing from predictions by our **coref-xlnet** model.

Next, we looked at examples for which our model predicted the mention-clusters incorrectly (affecting precision score). A subset of such examples in Table 6 shows that the most common mistake our model makes is incorrectly grouping related but distinct entities into the same cluster.

Original Example	**Predicted Clusters**	**Mistake**
[((41, 42), 'their'), ((36, 38), 'the investigators')] **[**((58, 59), 'they'), ((52, 54), 'The suspects'), ((0, 13), "The two main suspects in the bombing of the ` ` USS Cole ' '")**]**	[((0, 13), 'the two main suspects in the bombing of the the ` ` USS Cole ' '), ((36, 38), 'the investigators'), ((41, 42), 'their'), ((52, 54), 'the suspects'), ((58, 59), 'they')]	Two separate clusters grouped into one
[((255, 260), 'an SMS like this one'), ((284, 285), 'it')**]** **[**((304, 306), 'an SMS'), ((317, 318), 'it')**]**	[((258, 260), 'this one'), ((284, 285), 'it'), ((294, 295), 'it')]	Two separate clusters grouped into one

Table 6: Sample of example clusters that are predicted incorrectly by the **coref-xlnet** model.

We were also interested in examining the cluster predictions that were missed by both our model and the SpanBert-Base based coref model. This study (Table 7) revealed that both the models struggle to resolve coreferences for examples having apostrophe or quotation marks in them, and to also match phrases having similar semantic meaning. In addition, both the models find it hard to resolve pronoun coreferences.

[('I'), ('He'), ("Michael Wolf , a contributing editor for ` ` New York ' ' magazine , a media columnist , an important article in the new issue of ` ` New York"), ('Michael'), ('I'), ('you')]	Examples having apostrophe or quotes in them
[("the mainland ' s ' one China ' proposition"), ("' one China '")]	
[('my interview with Tom Friedman'), ('my conversation with New York Times columnist Tom Friedman about his book the new online charge to read him and Judy Miller')]	Failure to match phrases having similar semantic meaning
[('the current administration'), ('the white house')]	

[('all this'), ('the underlying technology that is flattening the world')]	Failure to match pronouns with referenced phrases
[('that'), ('all these other distractions'), ('several things happened basically in the last three years'), ('it')]	

Table 7: Sample of example clusters that are missing from predictions by our **coref-xlnet** model. These predictions are missed by **coref** too.

Strengths: Finally, we examined examples where **coref-xlnet** performs better than **coref.** We found that most of the coreferences resolved by coref-xlnet but missed by coref involved **longer mention spans** (First 3 examples in Table 8). We also discovered our model to perform better in some cases of **pronoun resolution** as well (last 2 examples in Table 8).

Original Example	**Predicted Cluster**
[((298, 308), 'the innovators and entrepreneurs who actually put all this together'), ((346, 361), 'the innovators who are doing all of these things who are actually reshaping the world')]	[((298, 308), 'the innovators and entrepreneurs who actually put all this together'), ((346, 361), 'the innovators who are doing all of these things who are actually reshaping the world')]
[((128, 137), 'cultural riches on the spiritual plane and material plane'), ((139, 141), 'these riches')]	[((128, 137), 'cultural riches on the spiritual plane and material plane'), ((139, 141), 'these riches')]
[((8, 44), "an agreement in principle calling for HOFI North America Inc . to combine its North American cement holdings with Ideal in a transaction that will leave Ideal ' s minority shareholders with 12 . 8 % of the combined company"), ((126, 128), 'the agreement')] [((153, 155), 'The transaction'), ((173, 175), 'the transaction')],	[((28, 44), "a transaction that will leave ideal ' s minority shareholders with 12 . 8 % of the combined company"), ((153, 155), 'the transaction'), ((173, 175), 'the transaction')]
[((470, 475), 'the pre - amplifier of my stereo'), ((500, 505), 'the thing that they * * stoled * *'), ((479, 480), 'it'), ((485, 486), 'it')],	[((470, 475), 'the pre - amplifier of my stereo'), ((479, 480), 'it'), ((485, 486), 'it'), ((500, 505), 'the thing that they * * stoled * *')]
[((290, 291), 'them'), ((284, 291), 'sheep without a shepherd to lead them')]	[((284, 291), 'sheep without a shepherd to lead them'), ((290, 291), 'them')]

Table 8: Coreferences resolved by **coref-xlnet** but not predicted by **coref**. Although some of these predictions made by coref-xlnet are grouped incorrectly but these mentions are not even identified by coref.

11. Conclusion

We apply the XLNet-Base model to the coreference resolution task, and investigate it using the independent method of splitting documents into independent segments. Our best model shows an F1 score of 72.47% on the CoNLL 2012 dataset. We show that the performance improves as larger contexts are provided to the model. With our experiments, we also observe that models based on XLNet need to be trained longer when compared to other pretrained models, such as BERT. Finally, and most importantly, we note that using longer sequences and larger batches with XLNet greatly improves the model's performance and provides evidence that this approach is particularly useful for modelling longer document-level context, while understanding that these benefits can be difficult to realize because of the massive compute and memory requirements to use XLNet effectively.

12. References

[1] Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, *8*, 64-77.

[2] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5753-5763).

[3] Pradhan, S., Moschitti, A., Xue, N., Ng, H. T., Björkelund, A., Uryupina, O., ... & Zhong, Z. (2013, August). Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* (pp. 143-152).

[4]Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., & Xue, N. (2011, June). Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task* (pp. 1-27).

[5] Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., & Zhang, Y. (2012, July). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL-Shared Task* (pp. 1-40).

[6] Joshi, M., Levy, O., Weld, D. S., & Zettlemoyer, L. (2019). BERT for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*.

[7] Lee, K., He, L., Lewis, M., & Zettlemoyer, L. (2017). End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

[8] Lee, K., He, L., & Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Appendix

Overview of coref

From the paper Joshi et al. (2019b), for each mention span x, the model learns a distribution P(·) over possible antecedent spans Y:

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

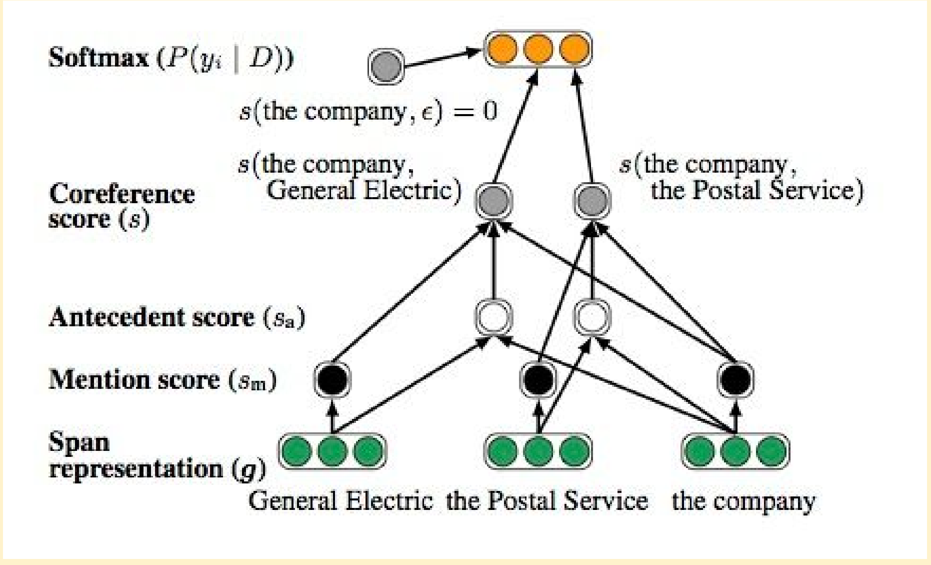
The span pair scoring function $s(x,y)$ is a feed-forward neural network over fixed-length span representations and hand-engineered features over x and y:

$$s(x,y) = s_m(x) + s_m(y) + s_c(x,y)$$

$$s_m(x) = FFNN_m(g_{\mathbf{x}})$$

$$s_c(x,y) = FFNN_c(\mathbf{g_x}, \mathbf{g_y}, \varphi(x,y))$$

Here g_x and g_y denote the span representations, which are a concatenation of the two transformer output states of the span endpoints and an attention vector computed over the output representations of tokens in the span. S_m represents the mention score of a span, and $s_c(x,y)$ is the coreference score for y being an antecedent of x. $FFNN_m$ and $FFNN_c$ represent two feedforward neural networks with one hidden layer, and $\varphi(x,y)$ represents the hand-engineered features (e.g. speaker and genre information). A more detailed description of the model can be found in Joshi et al. (2019b).


Softmax ($P(y_i \mid D)$)
s(the company, ϵ) = 0
s(the company, General Electric)
s(the company, the Postal Service)
Coreference score (s)
Antecedent score (s_a)
Mention score (s_m)
Span representation (g)
General Electric
the Postal Service
the company

Source: e2e-coref: End-to-end Neural Coreference Resolution

The antecedent score S_a of two spans in the above diagram means the same as S_c in our mathematical expressions above.