D-Coref: A Fast and Lightweight Coreference Resolution Model using DistilBERT

Chanchal Suman[1], Jeetu Kumar[2], Sriparna Saha[1], and Pushpak Bhattacharyya[1]

[1]Department of Computer Science & Engineering, Indian Institute of Technology Patna, India
email: {1821cs11, sriparna}@iitp.ac.in, pushpakbh@gmail.com
[2]Department of Computer Science, RKMVERI, Belur Math, Howrah, India
email: 0jkpandey@gmail.com

Abstract

Smart devices are often deployed in some edge-devices, which require quality solutions in limited amount of memory usage. In most of the user-interaction based smart devices, coreference resolution is often required. Keeping this in view, we have developed a fast and lightweight coreference resolution model which meets the minimum memory requirement and converges faster. In order to generate the embeddings for solving the task of coreference resolution, DistilBERT, a light weight BERT module is utilized. DistilBERT consumes less memory (only 60% of memory in comparison to BERT-based heavy model) and it is suitable for deployment in edge devices. DistilBERT embedding helps in 60% faster convergence with an accuracy compromise of 2.59%, and 6.49% with respect to its base model and current state-of-the-art, respectively.

1 Introduction

Edge devices require natural language processing (NLP) for understanding users' input[1]. Whenever it comes to user interaction, coreference resolution becomes an important task for analysing user's input. It involves determining all referring expressions that point to the same real-world entity. A grouping of referring expressions with the same referent is called a coreference chain or cluster. The goal of a coreference resolution system is to output all the coreference chains of a given text (Martschat and Strube, 2015; Ferreira Cruz et al., 2020).

Several works on coreference resolution are available in the literature having very high accuracy (Lee et al., 2017, 2018; Kantor and Globerson, 2019; Joshi et al., 2019; Fei et al., 2019). These models use ELMo (Lee et al., 2018) and BERT (Joshi et al., 2019) for learning the semantic space of the input. Because of the use of such heavy transformers with millions of parameters, these models require a lot of memory. However, smart devices like smartphones should be responsive, light-weight, and energy-efficient models. This motivates us to design a light-weighted coreference resolution model which is suitable for the deployment in smart-devices.

We contributed in word context representation of c2f-model (Lee et al., 2018), by forming it from embedding generated by DistilBERT instead of ELMo. We use c2f-model as our baseline model, since this is used as base model in all the recent works ((Kantor and Globerson, 2019), (Joshi et al., 2019)). DistilBERT is a smaller, faster, cheaper, and light-weight distilled version of BERT, which is approx. 97% efficient in comparison to BERT. It is 40% smaller in size, and 60% faster (Sanh et al., 2019). The embeddings are generated from the DistilBERT for learning the semantic space of the sentences. After the generation of embeddings, they are passed to the bidirectional LSTM, followed by span head for calculation of mention scores. These mention scores are used for forming the coreference chain using hierarchical clustering as defined in (Lee et al., 2018).

The standard CoNLL-2012 (Pradhan et al., 2012) dataset is utilized for the performance evaluation of our proposed model. Experimental results show that, the developed system requires only 60% memory for execution, in comparison to the BERT-based heavy models, while remaining 97% efficient, and 60% faster converging too. We have also shown that 768 embedding dimension is sufficient for word context embedding generation from DistilBERT.

[1] https://www.iotforall.com/iot-natural-language-processing/

Proceedings of the 17th International Conference on Natural Language Processing, pages 323–328
Patna, India, December 18 - 21, 2020. ©2020 NLP Association of India (NLPAI)

2 The Proposed Approach

In order to utilize the embeddings generated by DistilBERT for extracting the coreference chains, we have integrated the recently proposed higher-order coreference model proposed in (Lee et al., 2018) in our system. We refer to this work as c2f-model.

2.1 Overview of c2f-model

For each mention span u, the model learns a distribution $P(\cdot)$ over possible antecedent spans v, as shown in equation 1. The scoring function s(u,v) between spans u and v takes g_u and g_v as its inputs. It uses fixed-length span representations. The scoring function consists of a concatenation of three vectors: the LSTM states of both the span end-points and an attention vector computed over those span tokens. The score $s(u,v)$ is computed by the mention score of u ($s_m(u)$), mention score of v ($s_m(v)$), the joint compatibility score ($s_c(u,v)$) of u and v. The mention score of a span signifies the probability of a span to be a mention. The joint compatibility score signifies the probability of the two spans as corefering. The components are computed as follows:

$$P(v) = \frac{e^{s(u,v)}}{\sum_{v' \in V} e^{s(u,v')}} \tag{1}$$

$$s(u,v) = s_m(u) + s_m(v) + s_c(u,v) \tag{2}$$

$$s_m(u) = FFNN_m(g_u) \tag{3}$$

$$s_c(u,v) = FFNN_c(g_u, g_v, \phi(u,v)) \tag{4}$$

where $FFNN(\cdot)$ represents a feed forward neural network and $\phi(u,v)$ represents speaker and meta-data features. Antecedent distribution is used for further refinement of these generated span representations. Finally coreference chain is formed using the scores generated from the softmax layer.

2.2 Extraction of Embedding from DistilBERT for word context representation

Extraction of embeddings from ELMO is shown in (Peters et al., 2018). We have shown the embedding extraction from DistilBERT for word representation in Fig. 1. This extraction of word representation is performed in 4 steps, which are explained below.

Conversion of Term-Tokens into WordPiece tokens: DistilBERT takes token embedding and position embeddings as input (Sanh et al., 2019). Thus, complete sentences are formed from the Term-Tokens, and passed to the DistilBERT tokenizer. DistilBERT takes WordPiece tokens generated by the tokenizer and merges the initial embeddings and the position embeddings as the final input for it.

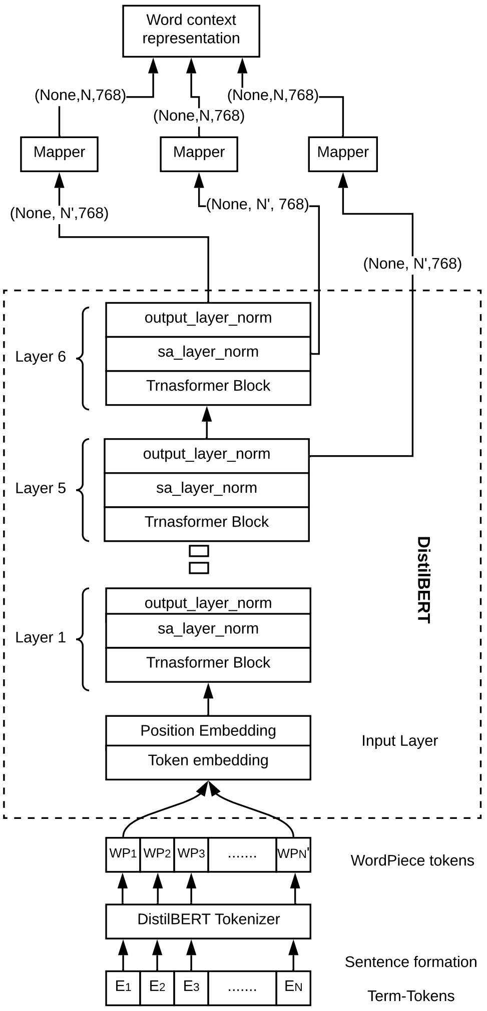

Figure 1: Word context representation from DistilBERT

Collecting outputs from DistilBERT: After, the generation of WordPiece tokens, these are given as input to the DistilBERT for the generation of word embeddings. For generating the word context representations from the ELMo, the three features 1) output of left-LSTM, 2) output of right-LSTM and 3) final embedding have been considered in the c2f-model. Similarly, to strengthen the learning from word context representation we generate word context representation from triplet of embedding outputs. We consider the raw form of embedding outputs from sa_layer_norm of Layer-6 and output_layer_norm of Layer-5 and final embedding output from the output_layer_norm of Layer-6 of DistilBERT. Word context representation means representation of word in the input sentence. *Word representation* as defined in c2f-model, are generated by character embedding using GloVe(Pennington et al., 2014) vector.

Mapping Embedding from WordPiece tokens to Term-tokens : The dimension of embedding matrix generated from DistilBERT is $(None, N', 786)$. Here, N' is the maximum of the number of WordPice tokens for a sample point in the batch. Learning the coreference in context of WordPiece token is complex to understand, and its analysis and explanation seem unusual. So we have mapped the output of WordPiece token to Term-Token by averaging the corresponding WordPiece embeddings.

Let, in a batch of size B, N be the maximum of number of Term-Tokens in a sample, $N^{'}$ be the maximum number of WordPiece tokens, and $i, j, k, l \in \mathbf{N}$. Let, WordPiece token generated by DistilBERT tokenizer be $WPT = \langle WP_1, WP_2, WP_3, \ldots, WP_{N'} \rangle$ (when the number of tokens in WordPiece token is less than $N^{'}$, then post-padding is done to get it) for input Term-Token, $ET = \langle E_1, E_2, E_3, \ldots, E_N \rangle$. Let, the embedding output generated from DistilBERT be $EmbOut^{'}$, which is a matrix of order $(B, N^{'}, 768)$, where

$$EmbOut^{'}[i] = \left[e^{'}_{j,k}\right]_{1 \leq j \leq N';\ 1 \leq k \leq 768};$$

$\forall 1 \leq i \leq B$ Then, we map it to the $EmbOut$ matrix of order $(B, N, 768)$ *i.e.*,

$$EmbOut[i] = [e_{j,k}]_{1 \leq j \leq N;\ 1 \leq k \leq 768};$$

$\forall 1 \leq i \leq B$ where

$$e_{j,k} = e^{'}_{j,k}; \tag{5}$$

$$\text{if } WP_j = E_j \times \frac{1}{l} \sum_{p=1}^{l} e^{'}_{j+p,k}; \tag{6}$$

$$\&\text{if } \Psi(j, l) = True \tag{7}$$

$\forall\ 1 \leq k \leq 768$ and the function $\Psi(j, l)$ *returns True* if the WordPiece tokens $\langle WP_{j+1}, \ldots, WP_{j+l} \rangle$ lead to term-token, E_j. Similar procedure is followed to get the embedding output from the rest of the two layers.

Formation of the final word context representation: The output from Layer-6, sa_layer_norm of Layer-6, and output of Layer_5 are separately passed to mapper and mapped output $m1, m2, and, m3$ are collected. Finally, $m1, m2, and, m3$ are concatenated for generation of the word context representation. This mapping also reduces the second dimension of word context representation from N' to N. Thus, the order of word context representation becomes $(None, N, 2304)$, where N is the maximum number of tokens in a sample in the batch; this reduction makes the model to work with less space too.

2.3 Overview of the proposed system

The word and character embeddings are generated via DistilBERT and Glove, respectively. The word embedding generation through DistilBERT is discussed in the subsection 2.2. Character embeddings are generated through Glove similar to the c2f-model. The flow of our model after embedding generation is same as that of the c2f-model. The embeddings are fed to bidirectional LSTM to learn encoded representations for the words. The encoded features are further passed ahead to form the span head and span representation with span head feature. These span representations are then used for calculating the coreference score. Mention score and antecedant score are used for calculating the final coreference score. The formula for these calculations is shown in the Equation 1. For determining the final probability distribution between different spans, softmax is applied. At last, hierarchical clustering is used to form the coreference chain using the generated probability distribution.

3 Dataset used and experimental set-up

CoNLL-2012 shared task corpus is a standard coreference resolution corpus (Pradhan et al., 2012). We have used the English-based corpus for evaluating the performance of our proposed approach.

Our experimental setup is almost similar to that of c2f-model and we have modified some parts of their code to generate word context representation from DistilBERT embeddings, which are:

1) The ELMo embeddings are replaced by the DistilBERT embeddings which are lighter and faster.

2) We have experimented with *word context representation*, generated from DistilBERT. The two different experimental setups are discussed below:

i) D-Coref-Small: In our proposed D-coref model, we have extracted the embeddings from the three layers of DistilBERT for generating the word context representation. The order of generated embedding is $(None, N, 768)$ and the order of word context representation is $(None, N, 2304)$.

ii) D-Coref-Large: For higher dimensional word context representation, we have extracted the embedding outputs from layer-4 of DistilBERT for raw embedding representation in addition with embeddings of D-coref-Small. Thus, the order of word context representation for this setup is $(None, N, 3072)$ similar to c2f-model.

Table 1: Comparison with previous works

	MUC	B^3	CEAF	Avg F1
(Lee et al., 2017)	75.8	65.0	60.8	67.2
(Lee et al., 2018)	80.4	70.8	67.6	73.0
(Joshi et al., 2019)	83.5	75.3	71.9	76.9
D-coref-Large	78.15	67.94	64.76	**70.28**
D-coref-Small	78.27	68.09	64.87	**70.41**

4 Results and Analysis

In this section, we have discussed the performance of our model on the standard CoNLL-2012 dataset, along with different features of the model.

4.1 Performance Evaluation

We have reported precision, recall and F1-scores of the B^3, MUC, and,CEAF metrics, and average F1 score (main evaluation metric) of all these three metrics as per the previous papers (Pradhan et al., 2012). The results obtained from our proposed approach is tabulated in table 1, and the detailed comparison is shown in table 2. Our baseline is the c2f-model with ELMo input features, which achieves an average F1 of 73.0%. We have achieved an average F1 of 70.41% for D-Coref-Small, and 70.28% for D-Coref-Large. Our experiments show that getting word context representation in the dimension of $(None, N, 2304)$ is sufficient. After observing the performance and the size of the model, we consider the D-Coref-small as our final model. The performance of D-Coref-small is 6.49% less than the current state-of-the-art (Joshi et al., 2019) and 2.59% less than the c2f-model. This performance matches with the claim of 3% less language understanding capability of the DistilBERT model[2]. We have a loss of approx 6% in performance, but this is the inherent nature of DistilBERT. At the same time, our model has become faster and lightweight due to the usage of the faster and lighter DistilBERT.

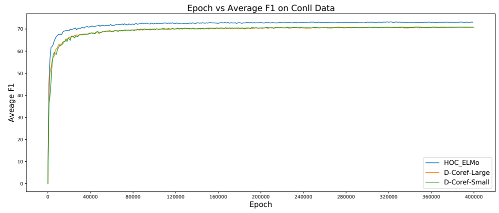

Figure 2: Epoch versus Average F1 curve

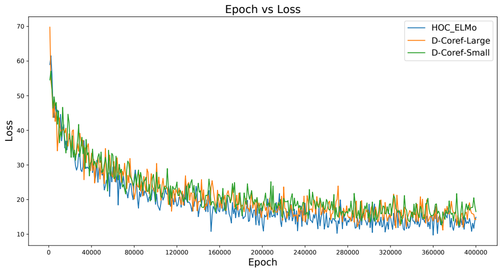

Figure 3: Epoch versus Loss curve

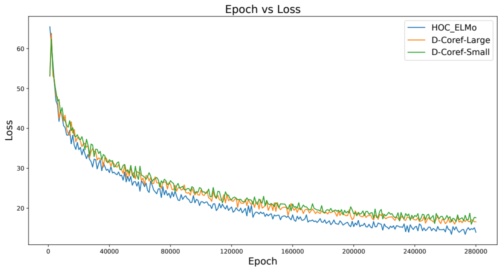

Figure 4: Epoch versus Average loss graph

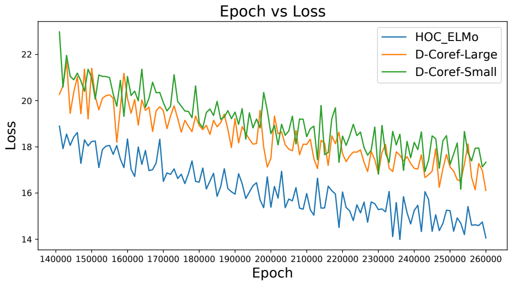

Figure 5: Epoch versus Average Loss sub graph

The detailed comparison table, with all the performance metrics are shown in Table 2. The epoch vs loss and epoch vs average F1 curves are shown in Figures 2, 3, 4, and 5.

[2] https://medium.com/huggingface/distilbert-8cf3380435b5

Table 2: Comparison with previous works

	MUC			B^3			CEAF			
	P	R	F1	P	R	F1	P	R	F1	Avg F1
(Martschat and Strube, 2015)	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
(Clark and Manning, 2015)	76.1	69.4	72.6	65.6	56.0	60.4	59.4	53.0	56.0	63.0
(Wiseman et al., 2015)	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
(Wiseman et al., 2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
(Clark and Manning, 2016)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
(Lee et al., 2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
(Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
(Joshi et al., 2019)	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
D-coref-Large	80.45	75.97	78.15	71.32	64.86	67.94	66.59	63.03	64.76	**70.28**
D-coref-Small	80.85	75.86	78.27	71.91	64.65	68.09	62.69	67.2	64.87	**70.41**

4.2 Characteristics df the Proposed Model

Our proposed model is fast and light-weight. Here, we have discussed these two properties in detail.

Fast: From the epoch vs loss graph (fig. 3), we observed that model does not show any improvement after 240K. But after examining the average F1 plot (fig. 2), it is evident that the model has converged at 200K and there is no improvement in average F1 after 200K, while the c2f-model had converged at 400K epochs. In this way, the developed model is 60% faster, this behaviour also matches with the claim of faster learning capability of the DistilBERT (Sanh et al., 2019).

Light-weight: DistilBERT is a very light-weight model, with 66 millions of parameters, while the transformer ELMo has 465 millions of parameters (Sanh et al., 2019). Thus it can meet the memory requirements of edge devices. The requirement of fewer parameters for DistilBERT is the main motivation of this work. At the same time, the reduced word context representation dimension of D-coref-small has also lowered the model size, because the entire learning dimension depends on word context representation as it flows throughout the model.

In the view of these advantages, it is evident that our model is suitable for small devices with some compromise in performance. The size of the DistilBERT is reduced by 40% in comparison to BERT model, while it retains 97% of the language understanding capabilities of BERT and is 60% faster (Sanh et al., 2019). Thus, the usage of DistilBERT embeddings makes our model faster and lighter.

5 Conclusion and Future Work

We have devised a fast and light-weight coreference resolution model using DistilBERT. In order to generate a faster and light-weight model, the accuracy gets compromised. Word context representation in reasonable lower dimension can work like representation in higher dimension with proper tuning. Our developed system requires only 60% memory for execution, in comparison to the BERT-based heavy models, while remaining 97% efficient too. Thus, it is suitable for edge devices. In future we will try to come up with a model having better performance with same or lesser space requirement.

Acknowledgments

Dr. Sriparna Saha gratefully acknowledges the Young Faculty Research Fellowship (YFRF) Award, supported by Visvesvaraya Ph.D. Scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research.

References

Kevin Clark and Christopher D Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415.

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

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665.

André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. 2020. Coreference resolution: Toward end-to-end and cross-lingual systems. *Information*, 11(2):74.

Mandar Joshi, Omer Levy, Daniel S Weld, and Luke Zettlemoyer. 2019. Bert for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*.

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.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418.

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

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sam Wiseman, Alexander M Rush, and Stuart M Shieber. 2016. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.

Sam Joshua Wiseman, Alexander Matthew Rush, Stuart Merrill Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.