CS 7650: Assessing the Impact of Embedding Document Quality for Coreference Resolution using BERT

Bipin Koirala, Paul Hutchison, Benjamin Colton

1 INTRODUCTION

Coreference resolution is the task of identifying expressions that refer to the same entity within or across documents and is a fundamental challenge in natural language understanding. Recent advances such as Cross-Document Language Modeling (CDLM) [1] have shown promising results in learning and identifying relationships between coreferences in disparate documents. Leveraging long-range transformers capable of modeling related documents jointly enables richer context modeling, which is the ability to understand and represent the broader linguistic, semantic, and referential context surrounding coreference mentions across multiple documents.

This project was initially motivated by the goal of fine-tuning a pretrained CDLM model for cross document coreference resolution, with a focus on integrating *document quality* as an additional signal during training. However, due to the substantial computation requirements of CDLM, especially with inputs length up to 4096 tokens, we encountered practical constraints such as limited training time and available computer memory that constrained our ability to prototype effectively. To address these challenges, we shifted our focus towards a more computationally tractable setup. Specifically, we fine-tuned a BERT-based model using Low-Rank Adaptation (LoRA) [5] and added a lightweight classifier head to predict coreference links. While this shift meant restricting one model to a *single-document* coreference analysis, we retained our original objective of exploring how document-level quality annotations might enhance model performance under limited-resource settings.

2 RELATED WORK

Early approaches to coreference resolution relied on heuristic and rule-based methods, later evolving into statistical models such as mention-pair and entity-mention models [9], [10]. With the advent of deep learning, span-based models leveraged contextual embeddings from pretrained language models (e.g. ELMo, BERT) and demonstrated significant improvements to single-document coreference tasks [3], [6], [7]. To extend coreference resolution beyond document boundaries, CDLM [1] introduced cross-document modeling. This method jointly models sets of related documents using Longformers with a dynamic global attention mechanism, allowing the model to learn inter-document dependencies during pretraining. CDLM yields strong performance across tasks such as multihop QA, document matching, and cross-document coreference.

Despite its success, CDLM is computationally expensive due to its reliance on long-sequence transformers and joint document encoding. Our project builds on this foundation but adapts the problem to a resource-constrained setting. Specifically, we adopt LoRA [5] as a means to efficiently fine-tune BERT [2]. LoRA injects low-rank trainable parameters into transformer layers, allowing for adaptation with minimal computational overhead, making it well-suited for rapid iteration.

3 METHODOLOGY

3.1 Dataset

We use the English subset of the *OntoNotes 5.0* corpus available at CoNLL-2012 Shared Task, which is also accessible through *HuggingFace* datasets library. The dataset provides annotated coreference chains over a wide range of documents from multiple genres, including Newswire, broadcast news, and conversational speech. Each document consists of pre-tokenized sentences along with mention spans and corresponding cluster annotations. We use the *train*, *validation*, and *test* split as provided in the dataset.

The State of The Art (SoTA) model for this dataset is Maverick, and achieves an average F_1 score of **0.836** [8]. This model is much larger than what is implemented here and obtains multiple document coreferences, not just single-document as will be shown here. This model, like this architecture, is based on BERT; more specifically it is based on DeBERTa. This model has $504M$ parameters and takes fourteen hours to train.

3.2 Preprocessing

We extract *mention pairs* from coreference annotations to create binary classification examples indicating whether a pair of mentions refers to the same entity. To construct positive examples, we group mentions by their cluster ID and randomly sample up to three positive pairs per cluster using combinations of mentions. This capping ensures diversity without over-representing large clusters. For negative examples, we consider all pairs of mentions from different clusters and downsample them to match the number of positive pairs, which results in a balanced and compact dataset that is computationally efficient to train on. This is particularly crucial given limited resources.

We tokenize each mention pair into a fixed-length input sequence for the model. Similar to the CDLM, mentions are wrapped with a special tokens <m> and </m>, which are added to the tokenizer's vocabulary. The input sequence is constructed as:

```
[CLS] ... <m> mention1 </m> ... <m> mention2
</m> ... [SEP]
```

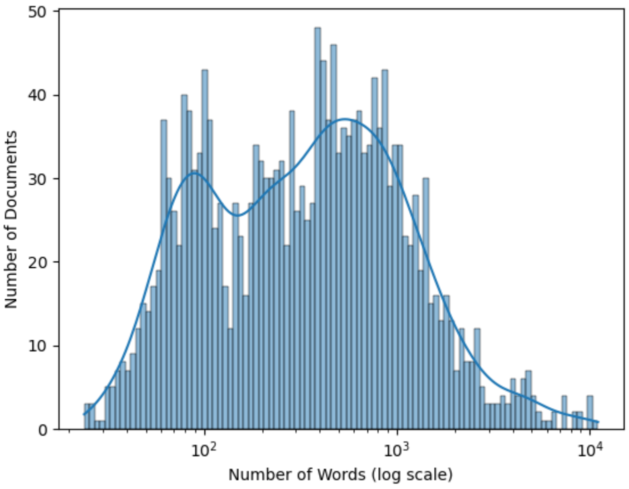

Figure 1: Histogram plot showing the total words per document across all training set. Considerable amount of documents exceed word count of 512 (BERT's maximum token length).

As depicted in Figure 1, the context around each mention is restricted to a window of 122 tokens to ensure that the resulting sequence fits within BERT's 512-token limit, including special tokens and both mentions. If the total tokenized length exceeds the limit, we iteratively trim the *between-mention* segment first, followed by *left* and *right* contexts.

Additionally, we encode *document quality* using three heuristic scores: S_1 (mention density) - proportion of tokens involved in coreference chains, S_2 (cluster richness) - average mentions per token, S_3 (length score) - normalized document length, i.e. $\min(\frac{\text{tokens}}{669.75}, 1)$, where 669.75 is the mean token count in the training set. Each score is discretized into one of four categories - low, mid, high, or very-high, based on $25^{th}, 50^{th}$ and 75^{th} quartile-based binnings - computed over the training split. In the final dataset, each sample is a plain-text with <m>-</m> tags surrounding the two mentions. If document quality encoding is enabled, special tokens such as <S1=high>, <S2=mid>, <S3=low> are prepended to the input sequence to condition the model on document-level attributes.

3.3 Model Architecture

We use BERT model *(bert-base-uncased)* as the core encoder. Instead of fine-tuning the entire model, we apply LoRA to inject a small number of trainable parameters into the attention layers of BERT. The LoRA configuration is as follows; $r = 8$ (rank of low-rank matrix), $\alpha = 8$ (scaling factor), dropout $= 0.1$ and target modules = ['query', 'value'] in the self-attention mechanism. For the baseline model, we extract the embeddings of [CLS], token span for mention1 and mention2 from the final hidden layer of BERT. Similar to CDLM, we then construct a concatenated tensor:

$$[\texttt{[CLS]}, m_1, m_2, m_1 \odot m_2]$$

where m_i represents the sum of embedding representation(s) for a given mention. This representation is then fed into dense feed-forward network, with ReLU activation, that acts as a classifier head for coreference resolution. For the document quality aware model, we also extract the embeddings corresponding to the scores and construct a different tensor:

$$[\texttt{[CLS]}, m_1, m_2, m_1 \odot m_2, S_1, S_2, S_3, S_1 + S_2 + S_3]$$

3.4 Training

To reduce the computational burden of fine-tuning LLM, we employ LoRA on the pre-trained BERT-base model, which consists of approximately 109.79 million parameters. With our current LoRA configuration, we reduce the number of trainable parameters to $294,912$, i.e., 99.74% reduction in parameter count compared to full fine-tuning.

We adopt the *AdamW* optimizer with a learning rate $\eta = 2 \times 10^{-5}$, to jointly fine-tune the LoRA augmented BERT and train the classifier head. Due to GPU memory limitations, we use a batch size of 1 for training. The model is trained for a maximum of 3 epochs, with early stopping triggered if no improvement in the validation F_1 score is observed after processing 4000 samples. The training objective is the *Binary Cross-Entropy Loss with logits*, defined as:

$$\mathcal{L}_{BCE}(\hat{y}, y) = -[y \cdot \log(\sigma(\hat{y})) + (1 - y) \cdot \log(1 - \sigma(\hat{y}))]$$

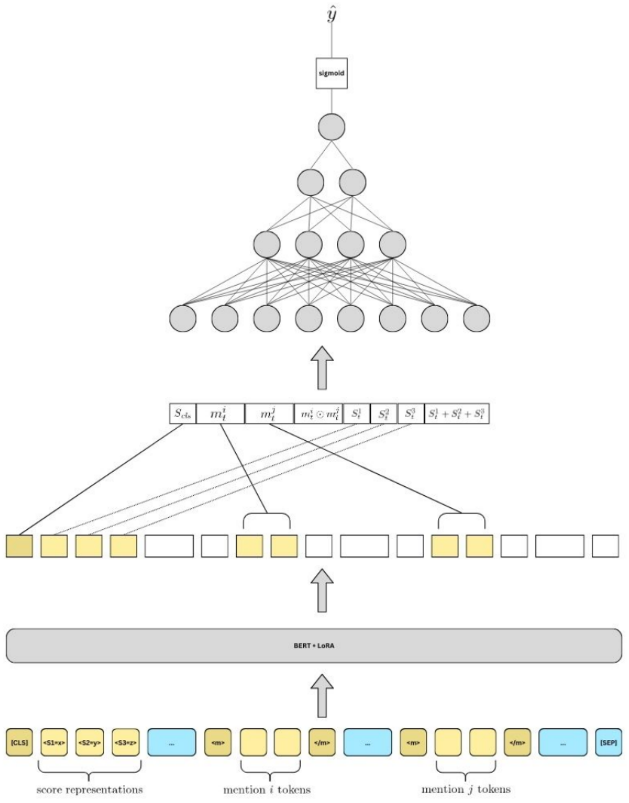

Figure 2: A representative diagram of document-score-aware model. In the Baseline model (same architecture but without LoRA), the three score tokens after `[CLS]` are not present and so as their corresponding embeddings.

where $\hat{y} \in \mathbb{R}$ is the predicted logit, $y \in \{0, 1\}$ is the ground truth label, and $\sigma(\hat{y}) = 1/(1+e^{-\hat{y}})$ is the sigmoid activation.

4 RESULTS

4.1 Findings

Table 1 presents a comparative evaluation of the Baseline model and the *document-quality-aware* (Scored) model for both the validation and test sets across 200 samples. The models are assessed across standard classification metrics: precision, recall, accuracy and F_1 score. Overall, the Baseline model consistently outperforms the Scored model across most metrics on both the validation and test splits. In particular, the Baseline achieves higher precision and F_1 score, suggesting that it is more effective at correctly identifying coreferent mention pairs without over-predicting. We also see that the baseline model has a comparable F_1 score to the SoTA model on this data set.

Metric	Baseline (Val)	Scored (Val)	Baseline (Test)	Scored (Test)
Precision	**0.8588**	0.7909	**0.8229**	0.7615
Recall	0.8111	**0.8365**	0.7980	**0.8557**
Accuracy	**0.8505**	0.7970	**0.8093**	0.7970
F1 Score	**0.8343**	0.8131	**0.8103**	0.8058

Table 1: Performance comparison of Baseline vs. Scored (Document-Quality aware) on Validation and Test sets

However, an interesting trend emerges in terms of recall. On both validation and test sets, the Scored model achieves higher recall compared to the Baseline model - **0.8365** vs. **0.8111** on the validation set, and **0.8557** vs. **0.7980** on the test sets. This indicates that the document-quality-aware model is more sensitive, successfully retrieving a greater number of true coreference links, although at the cost of reduced precision.

This trade-off reflects a classical precision-recall balance: while the Baseline model is more conservative in its predictions (favoring precision), the Scored model is more permissive (favoring recall). This behavior may be attributed to the additional signal introduced by the document quality annotations, which could be encouraging the model to identify more potential coreference pairs, including borderline or ambiguous cases.

Despite its lower $F1$ performance overall, the scored model's higher recall could be advantageous in downstream tasks where missing coreference links is more detrimental than introducing some false positives.

4.2 Analysis

To gain insight into the observed improvement in F_1 score of the documet-quality-aware model, we investigate the role of discrete score tokens, which encode document-level features such as the proportion of coreferent mentions, mention span density, and document length. These score tokens are inserted into the input sequence to provide external quality signals to the model during fine-tuning. To examine how these quality indicators interact with coreferent expressions, we analyze the attention weights from each score token to the tokens comprising the two mentions in a candidate pair. Specifically, we extract the attention values from the final layer of the transformer, averaged across all attention heads, and visualize the degree to which each score token attends to the tokens located between the `<m>` and `</m>` markers that surround *mention1* and *mention2* in the input sequence. This analysis enables us to observe whether and how document-level quality annotations influence the model's internal representation of coreferent mentions, potentially contributing to its improved recall and overall performance.

The attention heatmaps in Figure 3 reveal how document quality tokens influence the model's focus on mention tokens. In documents with a low proportion of coreferent mentions, i.e., `<S1=low>`, the model exhibits sharply peaked attention, specificially toward proper nouns (e.g. 'Beijing'), suggesting reliance on surface-level cues. In contrast, mid-and-high-quality score tokens distribute attention more evenly across the mention span, potentially enabling deeper contextual understanding. These patterns help explain the higher recall of the scored model; by attending more broadly in better-quality documents, it retrieves more coreference links, even if some are less precise.

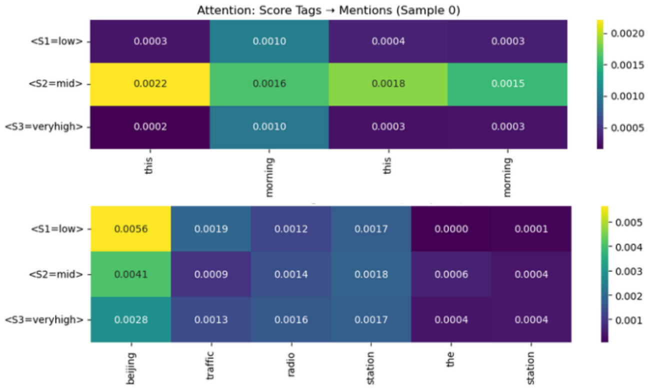

Figure 3: Attention from Document Quality Score Tokens to Mention Tokens. (Top): Both mentions are "this morning". (Bottom): The two mentions are "beijing traffic radio station" and "the station".

5 CONCLUSION

5.1 Summary

The primary result of this study is that embedding the document quality scores into the embedding in the way presented here did not improve the performance of the base model for coreference resolution. The method that is implemented did improve recall, which indicates that the document-quality-aware model is more successful in finding true coreferences, though will also be less precise. This may indicate that while this method did not explicitly improve the F_1 score over the base model, there may be ways to incorporate the document quality to better complete this task.

5.2 Limitations

While this work demonstrates the feasibility of incorporating document-level signals into mention-pair classification for coreference resolution, it also has several notable limitations. First, the three document-quality features are hand-crafted and coarse-grained heuristics. They may not accurately capture semantic richness or coherence, which are deeper and more impactful indications of a document's 'quality' from a linguistic standpoint, i.e., they are not grounded in deep linguistic theory or empirical quality judgments, limiting their effectiveness. Second, the major limitation of this effort was the size of the model, which limited the model to single-document coreference analysis, rather than longer coreference chains. It is possible that a larger context in a larger model could better capture the true coreferences and benefit from the document quality scores in the embedding. Third, due to limited resources for computation, we were unable to perform a thorough hyperparameters search to obtain the best possible results. Furthermore, our current setup lacks end-to-end architecture because the system is designed as a mention-pair classifier over pre-extracted mentions. This design assumes perfect mention detection, which is not realistic.

5.3 Future Works

Our findings suggest that incorporating document-quality information can improve recall in coreference resolution, although at the expense of precision and accuracy. This trade-off opens several promising directions for future exploration:

- **Gating mechanism:** Modulate the model's attention or feature weighting based on quality scores.
- **Model Ensembling:**[4] Baseline model outputs are used for high-precision predictions and Scored model contributes to identifying more borderline or uncertain coreference links (high recall).
- **Learnable Document Quality Scores:** Learned embeddings from a secondary model trained to predict 'coreference-friendliness' of a document.
- **Extend to Cross-Document:** Scale current architecture to handle cross-document coreference, where quality scores could help disambiguate low-context links.

REFERENCES

[1] Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. *arXiv preprint arXiv:2101.00406* (2021).

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[3] 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*. 660–665.

[4] Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence* 115 (2022), 105151.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[6] Tuan Manh Lai, Trung Bui, and Doo Soon Kim. 2022. End-to-end neural coreference resolution revisited: A simple yet effective baseline. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8147–8151.

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

[8] Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. 2024. Maverick: Efficient and Accurate Coreference Resolution Defying Recent Trends. arXiv:2407.21489 [cs.CL] https://arxiv.org/abs/2407.21489

[9] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* 27, 4 (Dec. 2001), 521–544.

[10] Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu, and Sheng Li. 2008. An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming. In *Proceedings of ACL-08: HLT*, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui (Eds.). Association for Computational Linguistics, Columbus, Ohio, 843–851. https://aclanthology.org/P08-1096/

A APPENDIX

A.1 Resources and Reproducibility

All data, trained models, and code artifacts used in this project are available at the following link:

[Access Models and Dataset].

This includes:

- Preprocessed dataset
- Fine-tuned Baseline BERT model + Classifier
- Fine-tuned Score-Aware BERT model + Classifier
- Scripts for data processing, training and evaluation