Enhancing Coreference Resolution through Integrating POS Tagging in Transformer Models

Ata Halıcıoğlu
ahalicioglu20@ku.edu.tr

Oğuz Kağan Hitit
ohitit20@ku.edu.tr

1 Problem statement

Coreference resolution is an important task in natural language processing (NLP) that focuses on identifying expressions within a text that refer to the same real-world entity. Despite the rapid advancements in NLP, fueled by increased computational power and the development of large-scale models that can be fine-tuned for specific tasks, coreference resolution continues to present significant challenges. These challenges arise primarily from the complexities of natural language, which include contextual ambiguities and diverse language structures. Each of these factors can dramatically shift the meaning of a text and complicate the task of linking referential phrases accurately.

The complexity of coreference resolution lies in its need to understand and interpret comprehensive texts, which often involve complex relationships between entities. For instance, the task must distinguish whether pronouns like ”he” or ”she” refer to previously mentioned individuals or new entities entirely. Additionally, it must handle cases where entities are referred to in various syntactic forms—such as direct mentions, pronouns, or descriptive noun phrases.

In our paper, we propose an innovative approach to improve coreference resolution by incorporating part-of-speech (POS) tagging into the process. POS tagging assigns words in a sentence to categories like nouns, verbs, and adjectives based on their definition and context—how they relate to nearby words. This integration aims to use the syntactic information provided by POS tags to enhance the identification and interpretation of different noun types and their semantic roles within sentences.

For instance, distinguishing between ”a city” and ”London” involves recognizing one as a common noun and the other as a proper noun, which is crucial for accurate reference resolution. Our hypothesis is that by embedding detailed syntactic data from POS tagging into coreference systems, these systems will gain a deeper understanding of complex sentence structures, therefore improving their ability to identify and link referents across a text.

2 What we proposed vs. what we accomplished

2.1 What we proposed

- **Training a custom cransformer**

We planned to develop and train a custom transformer model that was specialized in coreference resolution using POS tagging

- **Fine-tuning pretrained models with POS tagging**

We planned to fine-tune pretrained models, specifically BERT and SpanBERT, with integrating POS tagging information using OntoNotes v5.0 dataset used in CoNLL-2012 Shared Task.

- **Comparative Analysis**

Our goal was to conduct a comparison among our custom-trained transformer, and the fine-tuned versions of BERT and SpanBERT. We planned to assess their performance on coreference resolution tasks using the ConLL-2012 Shared Task benchmarks B^3, $CEAF_{\Phi4}$, and MUC, mentioned in the study of (Pradhan et al., 2012a).

2.2 What we accomplished

- **Fine-tuning BERT on multiple datasets**

We fine-tuned the base BERT model, not just on the OntoNotes v5.0 dataset as originally planned, but also on the GAP (Gender Ambiguous Pronouns) dataset introduced in the study of (Webster et al., 2018). We intended to test the model's performance across different types of coreference resolution data. Additionally, GAP dataset was easier to implement.

- **Comparative Evaluation**

We were not able to implement the originally planned B^3, $CEAF_{\Phi 4}$, and MUC metrics due to technical challenges. Therefore, we developed our evaluation strategies. We developed two alternative tasks to measure the performance of trained models: binary classification accuracy and span index prediction accuracy. We designed these metrics to evaluate the performance of the models in identifying correct referents and their positions within the text, respectively. We compared the performance of our baseline model against its fine-tuned versions using our metrics.

3 Related work

Coreference resolution, the task of identifying all expressions in text that refer to the same entity, has seen significant advances with the adoption of deep learning methods. Historically, the coreference resolution task was approached using a variety of methodologies ranging from rule-based systems to feature-driven machine learning models.

Initial approaches were predominantly rule-based, relying on manually crafted rules to identify coreference links based on syntactic and semantic cues (Hobbs, 1978), (Rich and LuperFoy, 1988). These systems were effective to a degree but also they were limited by their inability to generalize across different domains and they required extensive manual effort.

As machine learning techniques became more prevalent, researchers began to employ statistical models, which used features such as gender, number, and syntactic roles to predict coreferences. These feature-based models, including decision trees (McCarthy and Lehnert, 1995) and vector similarities with pre-determined features (Soon et al., 2001), (Ng and Cardie, 2002) offered similar accuracy over rule-based systems and were less labor-intensive to maintain.

With the advancements in deep learning, coreference resolution shifted significantly towards models that could learn complex patterns from data without extensive feature engineering. Pioneering works such as those by (Clark and Manning, 2016a), (Clark and Manning, 2016b) introduced neural network models that used mention-pair features to learn coreference probabilities, setting the stage for more advanced architectures.

The use of Recurrent Neural Networks (RNNs), such as RNN's, LSTM's (Long Short Term Memory) marked a crucial development in the coreference resolution task as well. RNNs, particularly those employing Long Short-Term Memory (LSTM) units, were utilized for their ability to process sequential data, maintaining a memory of previous inputs which is beneficial for tasks that require context over long text spans. This attribute made them particularly suitable for modeling the sequence of mentions and their potential references within a text (Wiseman et al., 2016), (Lee et al., 2017).

Building on the transformer architecture (Vaswani et al., 2023), the BERT (Bidirectional Encoder Representations from Transformers) model introduced by (Devlin et al., 2019) leveraged bidirectional training of transformers to better understand the context of a word based on all of its surroundings in a text. This capability proved highly effective in coreference resolution, leading to new state-of-the-art performances by enabling more nuanced understanding and representation of all possible mentions (Joshi et al., 2019), (Joshi et al., 2020).

There are also some approaches where they tried to integrate syntactic information in to the coreference resolution task, such as integrating linguistic information in to the pairwise scoring functions in the coreference calculation(Yang et al., 2023), (Otmazgin et al., 2023).

In our approach, instead of integrating the linguistic information into the scoring function, we will integrate it directly to the model.

4 Our Datasets

We used two widely recognized datasets: the OntoNotes v5.0 dataset for CoNLL-2012 Shared Task, and the GAP dataset.

- **OntoNotes v5.0**

The CoNLL-2012 Shared Task dataset, as introduced in the study of (Pradhan et al., 2012b), was developed to enhance multilingual coreference resolution capabilities. It includes data in three languages—Arabic, Chinese, and English. The dataset consists of approximately 13,100 documents, each annotated with both syntactic and semantic information, and is not limited to specific types of noun phrases or predefined entity categories which is important for detailed linguistic analysis.

In each document of the dataset, there are several annotated sentences including multiple fields such as coreference spans and POST tags along with several other fields such as named entities. Ontonotes v5.0 provides a list of all 51 POS-tags used in the dataset samples. Additionaly, there are 3 elements in each coreference span, the first one being the entity index, the latter being the start and end indices of the coreference span in the sentence.

Figure 1: Sample from Ontonotes v5.0 dataset

Category	**Content**
Words	WW, II, Landmarks, on, the, Great, Earth, of, China, :, Eternal, Memories, of, Taihang, Mountain
POS Tags	26, 26, 27, 18, 14, 26, 26, 18, 26, 5, 19, 27, 18, 26, 26
Coref. Spans	[59,8,8] (China) [74,13,14] (Taihang Mountain)

The primary challenge in using this dataset with the BERT model, which has a context size limit of 512 tokens, originated from the length of the documents. Each document in the dataset can contain more than 20 sentences, and coreference spans that refer to the same entity may exist across multiple sentences. To address this, we initially considered implementing a sliding window approach for training, where the model would receive loss signals and gradient updates after processing each document segment. However, due to the considerable length of the documents, this method was impractical to implement. Therefore, we decided to use a different method to work with the dataset, which will be explained later in the paper.

- **GAP (Gender Ambiguous Pronouns)**

The GAP dataset (Webster et al., 2018) is released by Google AI Language for the evaluation of the coreference resolution task. It is a gender balanced dataset, specifically created to avoid gender bias in coreference systems. With approximately 8,900 samples, this dataset is straightforward to use for models designed to predict if a candidate coreference span refers to the same entity as a reference coreference span. However, the dataset does not include any syntactic annotation like part-of-speech tags. To address this, we used the open-source BERT model of (Hassan Sajjad and Xu, 2022), which is fine-tuned for English POS tagging and achieves an F1-score of 96.69.

Figure 2: Sample from GAP dataset

Category	**Content**
Text	Her father was an Englishman “of rank and culture” and her mother was a free woman of color, described as light-skinned. When Mary was six, her mother sent her to Alexandria (then part of the District of Columbia) to attend school. Living with her aunt Mary Paine, Kelsey studied for about ten years.
Pronoun	her
A	Mary Paine
A-coref	FALSE
B	Kelsey
B-coref	TRUE

5 Our Baseline

Our baseline is the base BERT model, fine-tuned with coreference resolution data from the GAP and OntoNotes v5.0 datasets, separately. We chose the base BERT model due to its straightforward implementation using the Hugging Face Transformers library. As mentioned in Section 2, our initial plan was to develop a custom transformer for coreference resolution. However, we found that the available source codes on platforms such as GitHub were outdated and written in TensorFlow. This led us to use the BERT model from the Transformers library. Below, we detail the hyperparameters used for each dataset:

GAP Dataset

- **Learning Rate:** 1×10^{-5}
- **Epochs:** 5
- **Batch Size:** 32

- **POS-Tag Embedding Dimension:** 512

OntoNotes v5.0 Dataset

- **Learning Rate:** 1×10^{-5}
- **Epochs:** 1
- **Batch Size:** 32
- **POS-Tag Embedding Dimension:** 512

Both datasets come with predefined splits for training, validation, and testing, which we used in our experiments. We experimented with several learning rates (1e-5, 2e-5, 3e-5) and found that a learning rate of 1e-5 resulted in the lowest validation loss. We also tested various batch sizes (1, 32, 128, 512). Due to RAM limitations, larger batch sizes was impractical, and a batch size of 1 was slow, leading us to decide on batch size of 32. Regarding the number of epochs, the GAP dataset model converged after five epochs without significant improvement with additional training. On the other hand, with the OntoNotes v5.0 dataset, significant validation loss reductions started early in training and stabilized by the end of the first epoch. To avoid overfitting, we decided to train our model with a single epoch.

6 Our approach

Our approach involved integrating Part-of-Speech (POS) tags into the input sentences to enhance the model's understanding of each word's syntactic and semantic roles. Initially, we attempted to append the POS tags directly alongside the input text. However, this method was not successful due to the BERT model's input token limitation of 512 tokens. As described in Section 4, we attempted a sliding window approach, but it also did not resolve our issue due to us being unable to implement the approach in limited time properly.

Following many design ideas, we updated our model architecture to incorporate POS tag information into BERT's final hidden state by learning the embedding representations of the POS tags along the way during training. For training with the OntoNotes v5.0 dataset, we created an embedding layer with a vocabulary size of 51. Similarly, for the GAP dataset, we constructed an embedding layer with a vocabulary size of 46. These vocabulary sizes were determined by the number of POS tags available in the OntoNotes v5.0 dataset and the POS tagger used for the GAP dataset training, respectively. We used BERT's default tokenizer for both datasets. Our implementation involved the following steps:

1. Given input words and their corresponding POS tags, we feed the input words to BERT and their POS-Tags to the Embedding layer.
2. We concatenate the output of BERT and the vector embeddings of POS-Tags.
3. Using a classifier layer, we generate predicted outputs as probabilities that indicate whether the candidate coreference span refers to the same entity as the reference coreference span.

6.1 Mathematical model for training with GAP dataset

- Let x_i be the concatenation of tokens representing the context text and the entities under consideration for coreference, simplified with shorthand notations:

$$x_i = \text{[CLS]}\, s_i\, \text{[SEP]}\, r_i\, \text{[SEP]}\, c_i$$

  where s_i represents the entire text context, r_i represents the reference entity mention as pronoun, and c_i represents the coreference entity mention as noun, surrounded by special tokens [CLS] and [SEP].

- Let p_i represent the sequence of POS tags corresponding to each token in x_i, aligned token-by-token.
- Let e_{x_i} be the embedding obtained from $E(p_{x_i})$, where p_{x_i} are the POS tags for x_i.
- Let y_i be a binary variable associated with x_i, where $y_i = 1$ indicates that entities r_i and c_i co-refer to the same entity, and $y_i = 0$ indicates the absence of co-reference.
- Let h_i be the output from the BERT model for x_i.
- Let $\hat{y}_i$ be the predicted output, computed using a classifier layer based on h_i and e_i.

6.1.1 Training loop

$$h_i = \text{BERT}(x_i)$$

$$e_i = \text{E}(p_i)$$

$$z_i = x_i + e_i$$

$$\hat{y}_i = \sigma(\text{FFNN}(\text{ReLU}(\text{FFNN}(z_i))))$$

$$L_{CE} = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

6.2 Mathematical model for training with OntoNotes v5.0 dataset

- Let x_i be the concatenation of tokens representing two sentences and their corresponding coreference spans, simplified with shorthand notations:

  $$x_i = \text{[CLS]}\, s_{i1}\, \text{[SEP]}\, s_{i2}\, \text{[SEP]}\, c_{i1}\, \text{[SEP]}\, c_{i2}$$

  where s_{i1} and s_{i2} are the first and second sentences, respectively, and c_{i1} and c_{i2} represent the coreference spans associated with each sentence. [CLS] and [SEP] are special tokens of BERT.

- Let p_i represent the sequence of POS tags corresponding to each token in x_i, aligned token-by-token.

- Let e_{x_i} be the embedding obtained from $E(p_{x_i})$, where p_{x_i} are the POS tags for x_i.

- Let y_i be a binary variable associated with x_i and p_i, where $y_i = 1$ indicates c_{i1} and c_{i2} corefer to the same entity, and $y_i = 0$ indicates its absence of co-reference.

- Let h_i be the output from the BERT model for x_i.

- Let $\hat{y}_i$ be the predicted output, computed using a classifier layer based on h_i and e_i.

6.2.1 Training loop

$$h_i = \text{BERT}(x_i)$$

$$e_i = \text{E}(p_i)$$

$$z_i = x_i + e_i$$

$$\hat{y}_i = \sigma(\text{FFNN}(\text{ReLU}(\text{FFNN}(z_i))))$$

$$L_{CE} = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

In the OntoNotes v5.0 training setup, the dataset is not pre-structured for direct use in coreference resolution tasks. Consequently, we implemented negative sampling to create a balanced set of training samples suitable for the model. Our negative sampling approach works as follows: if $y_i = 1$, it indicates that the coreference spans from both sentences refer to the same entity within the document. In contrast, if $y_i = 0$, it means that while s_{i2} includes the same entity mentioned in s_{i1}, the coreference span in s_{i2} does not include that entity's span. This method of selective exclusion helps us generate inputs labeled as 0, enhancing the training process's effectiveness.

Additionally, the reason for choosing our classifier is to introduce non-linearity with the activation functions in order for model to understand more complex behaviour. The feedforward networks with the ReLU activation function achieves this. With the final activation function sigmoid, we get a probability score between 0 and 1 which corresponds to two entities referring to each other or not

Further, using Binary Cross-Entropy Loss (BCE) for our coreference resolution task provides several benefits ideal for binary classification. BCE is suitable for our our task where each pair of words must be classified as either referring to the same entity or not. This loss function interprets the model's outputs as the probability of the positive class, fitting with our goal of predicting whether two given words refer to the same entity. Moreover, this approach allows us adjusting the classification thresholds which is important for optimizing precision and recall metrics.

6.3 Our evaluation metrics

Our evaluation metrics focus on two key aspects of coreference resolution: binary classification for coreference identification and span index prediction. Each of these tasks uses different algorithms to assess the coreference resolution of our models.

6.3.1 Binary classification for coreference for the same entity

This approach evaluates whether a mention in the second sentence is the correct coreference of a mention in the first sentence. This binary classification task helps in verifying the accuracy of entity linking between two text spans.

6.3.2 Span index prediction for a given sentence

This approach involves predicting the most likely coreference span in the second sentence based on a mention in the first sentence. The approach is detailed below.

For each potential coreference span in the second sentence, generate a separate input instance:

$$x_{i1} = [CLS]\, \text{s}_{i1}\, [SEP]\, \text{s}_{i2}\, [SEP]\, \text{c}_{i1}\, [SEP]\, \text{c}_{i21}$$

$$x_{i2} = [CLS]\, \text{s}_{i1}\, [SEP]\, \text{s}_{i2}\, [SEP]\, \text{c}_{i1}\, [SEP]\, \text{c}_{i22}$$

$$x_{i3} = [CLS]\, \text{s}_{i1}\, [SEP]\, \text{s}_{i2}\, [SEP]\, \text{c}_{i1}\, [SEP]\, \text{c}_{i2c}$$

$$\vdots$$

$$x_{in} = [CLS]\,\mathrm{s}_{i1}\,[SEP]\,\mathrm{s}_{i2}\,[SEP]\,\mathrm{c}_{i1}\,[SEP]\,\mathrm{c}_{i2n}$$

For each x_i, the model assesses the likelihood of each span being the correct coreference. The span c_{i2c} represents the correct coreference span among the candidates. This evaluation method systematically assesses the model's ability to determine exact textual references.

7 Results

We assessed the performance of our models using GAP and OntoNotes v5.0 datasets. Our analysis includes results from binary classification task and span index prediction. We compared the performance of baseline models, which do not utilize POS-Tags, against our improved models that incorporate POS-Tag integration to demonstrate the enhancements achieved.

Table 1: Binary classification results for GAP

Model	**Acc.**	**Prec.**	**Rec.**	**F1**
Base	0.64	0.62	0.56	0.59
Enh.	0.75	0.84	0.54	0.66

Table 2: Binary classification results for OntoNotes v5.0

Model	**Acc.**	**Prec.**	**Rec.**	**F1**
Base	0.91	0.94	0.88	0.91
Enh.	0.92	0.93	0.91	0.92

Table 3: Span index prediction results for OntoNotes v5.0

Model	**Acc.**	**Prec.**	**Rec.**	**F1**
Base	0.80	0.16	0.82	0.27
Enh.	0.85	0.19	0.73	0.30

8 Discussion and Analysis

The GAP dataset is specifically structured to explore coreference relationships between a reference pronoun and various candidate nouns. Consequently, each sample within the dataset aims to build links exclusively between pronouns and nouns. Given this structure, we hypothesized that incorporating Part-of-Speech (POS) tags, which distinguish whether the candidates and references are nouns or pronouns, would enhance the effectiveness of coreference resolution. To test this hypothesis, we compared the F1 scores from our initial training (without POS tags) to the scores after integrating POS tags into our model. The results showed a noticeable improvement in performance, with the F1 score increasing from 0.59 to 0.66 in the binary classification task. This confirms the validity of our hypothesis, showing that the use of POS tags contributes positively to the model's ability to distinguish coreferential relationships more accurately.

In contrast to the GAP dataset, the OntoNotes v5.0 dataset includes coreferences of the same entities represented not only by nouns and pronouns but also by other parts of speech. This variety in POS tags complicates the model's learning process, particularly in identifying coreference links when the same entity is represented by different parts of speech. In the GAP dataset, where coreferences are limited to nouns and pronouns, the model encounters a more simplified learning process. This complexity is likely the reason why we observed a small increase in the F1 score for the binary classification task in the OntoNotes v5.0 dataset, moving from 0.91 to 0.92.

To support this observation, we hypothesize that if we were to refine the training data in the OntoNotes v5.0 dataset to include only noun-pronoun coreference links, similar to the configuration in the GAP dataset, we would likely observe a similar increase in the F1 score for binary classification tasks. This hypothesis is based on the assumption that limiting the scope of coreference links to nouns and pronouns will simplify the model's learning process and improve its performance, similar to the results obtained with the GAP dataset.

Additionally, our span index prediction algorithm introduced computational challenges. After reviewing the coreference spans in the OntoNotes v5.0 dataset, we noted that these spans typically consist of multiple words. Therefore, we initially planned for the model to predict the correct coreference span among all possible n-gram combinations of words within the candidate sentence s_{i2}. However, this approach significantly increased the number of samples in the test dataset for span index prediction, making it impractical for our use. Despite these challenges, our results for the OntoNotes v5.0 dataset showed an F1 score increase of 0.03 points, from 0.27 to 0.3. This indicates that our algorithm still offers valuable insights for further analysis.

Furthermore, we were unable to apply the span index prediction task to the GAP dataset due to its structure. As shown in Figure 2, each input sample of the GAP dataset consists of only one sentence or paragraph, limiting us from using two separate sentences for evaluation with span index prediction task.

9 Contributions of group members

As Ata and Oğuz, we worked closely on our project. We generally started by talking in detail about our plans. Then, we shared our first attempts with each other and gave feedback. This process of discussing and reviewing our work helped us improve our ideas before we finalized them. When we were together, how we worked depended on what the task needed. Sometimes we split up the work and focused on different parts by ourselves, and other times we did pair programming, where we coded together to solve problems more efficiently. This flexible way of working together made sure we used both of our skills and views effectively.

10 Conclusion

This project was both challenging and rewarding. The primary challenge was due to our limited prior experience with NLP research and techniques. Additionally, translating our conceptual ideas into practical implementations was difficult due to technological challenges such as improving processing efficiency and maximizing GPU utilization.

Despite the challenges, the project was very rewarding. Seeing our ideas work, which really helped our model perform better was a great feeling. During the research and setting up stages, we learned a lot about how NLP works and saw first-hand how our new ideas made a difference.

Looking ahead, we're excited to try different models using POS tagging. We would like to test out models like SpanBERT and sequence-to-sequence models such as T5. Also, it would be great if we could get better at handling longer documents by using a sliding window technique for processing the input.

11 AI Disclosure

- **Question**: How can i make ... that goes down in latex format

- **Answer**: In LaTeX, to create text that ”goes down” vertically, you can use the rotatebox or beginturn commands from the graphicx package, or arrange your text in a table or a parbox and rotate it. Here's how you can do this:

  Using rotatebox: This command allows you to rotate the text by any degree. To make text go downwards, you can rotate it by -90 degrees. Here's an example:

  documentclass article
  usepackagegraphicx
  begindocument
  rotatebox[origin=c]270This text goes down.
  enddocument
  Using beginturn:
  This is another method provided by the rotating package. It's similar to rotatebox but used within an environment. Here's how you can use it:

  documentclassarticle
  usepackagerotating

  begindocument
  beginturn270
  This text goes down.
  endturn
  enddocument
  Both methods will effectively rotate your text to go downwards. Choose the one that best fits your overall document setup. If you have other specific formatting needs or run into issues, feel free to ask!

- **Question**: No, what I mean is I have some examples such as a1, a2 where I want to write them line by line, but I need to write it n times, thus instead of writing them after two example I want to put ... and at last I want to give the last example, how can I do that

- **Answer**: To represent a sequence of examples in LaTeX where you begin with a few items, then show an ellipsis (to indicate continuation), and finally display the last item, you can use a simple list or mathematical environment depending on the context. Here's how you can structure this:

1. Using Text Mode: If your examples are more textual (like listing items), you can use an ‘enumerate‘ or ‘itemize‘ environment and manually put in the ellipsis. Here’s an example:

documentclassarticle
usepackageamsmath
begindocument

beginitemize
item a1
item a2
item dots
item an
enditemize

enddocument

Using Math Mode If your examples are mathematical in nature, you might prefer using an aligned math environment. Here’s how you can do it:

documentclassarticle
usepackageamsmath
begindocument

beginaligned
$a_1,$
$a_2,$
$vdots$
$a_n endaligned$

$enddocument$

This uses the ‘amsmath‘ package for better formatting of mathematical elements. The ‘vdots‘ command specifically creates vertical dots suitable for this purpose. You can choose either the ‘itemize‘ (for text) or ‘aligned‘ (for math) method based on the nature of your content.

If your examples need to be presented in a different style or if there is any other specific formatting you’re looking for, let me know!

- **Where we used**: We used AI in Section 6 where we mentioned the equations using the LaTeX syntax.
- **How was the AI:** Initially, the LaTeX syntax the AI generated did not work well. However, after providing a more detailed explanation in follow-up questions, we received the exact output we needed.

References

Clark, K. and Manning, C. D. (2016a). Deep reinforcement learning for mention-ranking coreference models. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Clark, K. and Manning, C. D. (2016b). Improving coreference resolution by learning entity-level distributed representations. In Erk, K. and Smith, N. A., editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Hassan Sajjad, Nadir Durrani, F. D. F. A. A. R. K. and Xu, J. (2022). Analyzing encoded concepts in transformer language models. In *North American Chapter of the Association of Computational Linguistics: Human Language Technologies (NAACL)*, NAACL ’22, Seattle.

Hobbs, J. (1978). Resolving pronoun references. *Lingua 44*, pages 311–338.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans.

Joshi, M., Levy, O., Zettlemoyer, L., and Weld, D. (2019). BERT for coreference resolution: Baselines and analysis. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

McCarthy, J. F. and Lehnert, W. G. (1995). Using decision trees for coreference resolution.

Ng, V. and Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Otmazgin, S., Cattan, A., and Goldberg, Y. (2023). Lingmess: Linguistically informed multi expert scorers for coreference resolution.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012a). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In Pradhan, S., Moschitti, A., and Xue, N., editors, *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012b). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *EMNLP-CoNLL Shared Task*.

Rich, E. and LuperFoy, S. (1988). An architecture for anaphora resolution. In *Second Conference on Applied Natural Language Processing*, pages 18–24, Austin, Texas, USA. Association for Computational Linguistics.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Webster, K., Recasens, M., Axelrod, V., and Baldridge, J. (2018). Mind the gap: A balanced corpus of gendered ambiguou. In *Transactions of the ACL*, page to appear.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. In Knight, K., Nenkova, A., and Rambow, O., editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Yang, X., Ye, W., Breitfeller, L., Yue, T., and Wang, W. (2023). Linguistically-inspired neural coreference resolution. *Adv. Artif. Intell. Mach. Learn.*, 3:1122–1134.