Capstone Project: Comparative Exploration of Coreference Resolution with Transformer Models

Aditya A. Landge
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
al2960@rit.edu

Advisor
Cecilia O. Alm, Ph.D.
College of Liberal Arts
Rochester Institute of Technology
Rochester, NY 14623

Abstract—**Coreference resolution is the task of automatically determining the chain of expressions in language that refers to the same entity or antecedent. In this project, we have explored transformer models, like the BERT model and its recent variants (e.g., ALBERT), and have adapted them for performing coreference resolution tasks on the GAP (Gender Ambiguous Pronoun) dataset and measured their performance. Moreover, a complete quantitative and qualitative analysis of the data was done to compare the performances of various transformer models, like vanilla transformer models, trained and fine-tuned BERT and its variant models, and models from literature surveys on languages demonstrating different pronominal complexity.**

I. Introduction

A. Understanding the Coreference Resolution Problem

Coreference resolution is the computational linguistic or natural language processing task that aims to determine the set of expressions that allude to the same entity (anaphora). Language users refer to the same entity using different noun phrases or deictic indexicals such as pronouns when speaking or writing. Such coreference phenomena can be thought of as chained hyperlinks in natural language. However, coreference can make it difficult for natural language reasoning models that aim to process language meaningfully. It is useful for solving various other tasks such as named-entity recognition, question answering, among others. [1]. The task has attracted broad interest for a while, yet it remains a substantial challenge for NLP systems.

Here is an example of a coreference chain:

1. Barack Obama nominated **Hillary Rodham Clinton** as his **secretary of state** on Monday. He chose **her** because **she** had foreign affairs experience as a former **First Lady**.

   Antecedant: Hillary Rodham Clinton
   Coreference chain: secretary of state, her, she, First Lady

At its fundamental level, the problem can be boiled down to identifying a candidate entity given a reference from a corpus. However, on the grand scale, coreference resolution can also highlight the biases in the corpus and the models. An example of a language model predicting words covered using the [MASK] token can be seen in examples 2 and 3 below. This example highlights some inherent biases in such language prediction models:

2. ”The nurse notified the patient that, [MASK] shift would be ending in an hour.”
   Predicted Result: ”her”
3. ”The nurse notified the patient that, [MASK] blood would be drawn in an hour.”
   Predicted Result: ”his”

The above examples are from Winograd Schema [2], which can be thought of as a Turing test for machine translation and is used to identify if the system can capture common sense. In this project, the Winograd Schema [2] acted as a litmus test to determine which model of the transformers to use for the coreference resolution task.

B. Understanding Transformers & BERT

One of the goals in this project was to understand the technical basis of BERT. [3] and other Transformers based models in general.

Transformers: Transformers utilize an attention mechanism to form a global contextualized numerical representation and establish relationships between input and output and include encoder layers followed by layers of decoders. The **encoder** encodes an English sentence into a contextualized numerical representation using the attention mechanism, and the contexts are propagated to each encoder layer one by one. The contextualized numerical representation from the final layer of the encoder is then directed to the first layer of the decoder. The **decoder** captures conditional probability for the contextualized numerical representation to understand what parts of the sentence are important. Hence, higher weights are assigned to more important words in the sentence, and lower weights are assigned to less important words of the sentence. For tasks like question answering and machine translation, this helps determine which key terms or words can be assigned to the sentences to make them meaningful and preserve the context of the sentence. A benefit of this architecture is that each embedded token representation interacts with every other

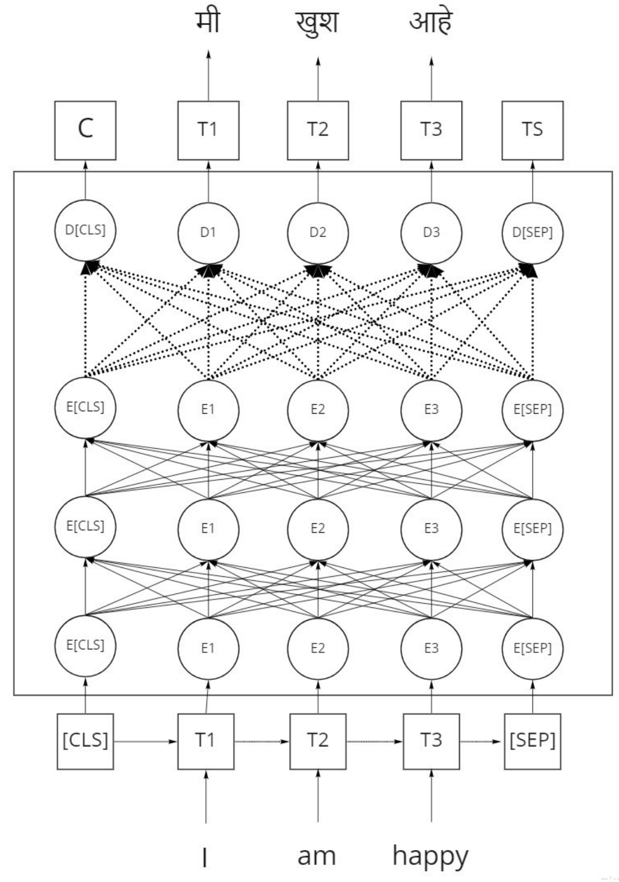

Fig. 1. This Figure shows a visual of the multi-layer transformer model. Transformer models typically have an equal number of encoder and decoder layers. In transformer models, an input sentence is initially converted into a list of numeric representations (tokenized). These tokenized numeric representations are then propagated to the encoder layers one by one. The output of the last encoder layer is propagated to the decoder layers one by one to obtain the contextualized numerical representation for the entire sentence. The representation in the final layer is used to build the output (a translated sentence in Marathi in this case)

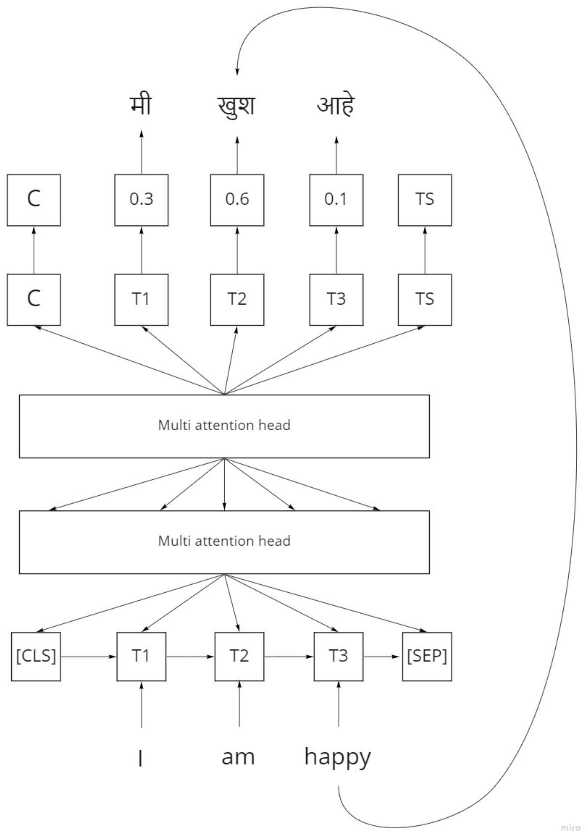

Fig. 2. This Figure provides a visual of a BERT model. The BERT model initially tokenizes the input sentence to form a numeric representation. Then the multi-head attention mechanism masks random tokens from different places within the sentence, and each attention head tried to predict the masked token in parallel. In the end, attention scores from all the attention heads are combined to form a contextualized numeric representation.

embedded token representation in the sentence to understand the context of the sentence. GPUs can be used for training.

As shown in Figure 1, the attention mechanism architecture supports learning how current word links to other words sequentially (e.g., anticipated lexical collocations and structural patterns). Transformers are known to decrease training times and improve results over RNNs[4].

BERT (Bidirectional Encoder Representations from Transformers) [3] and similar pre-trained language models have shown effective for improving performance on many NLP tasks. BERT is trained on language modeling tasks. In the training phase of BERT, around 15 % of random words in a sentence are covered with the [MASK] tokens, and the language models are used to predict these masked words. Since BERT reads entire sentences at a time, it preserves the context of all the surrounding words in the sentence and provides more accurate results predicting the [MASK] tokens. Python libraries like `transformers` provide various pre-trained models of BERT, who are trained on text corpora like Wikipedia and BookCorpus[5]. Some of the provided pre-trained models are `bert-base-uncased`, `bert-base-cased`, `bert-large-uncased`, `bert-base-multilingual-uncased`, etc. These pre-trained models provide a base that then enables transfer learning. These pre-trained BERT can be adapted, by adding another layer, to transfer the model to perform a variety of tasks.

A visual of BERT used for translating English to Marathi is in Figure 2. It learns contextual embedding between the source and target texts. BERT reads in the sequence of words and learns the context of surrounding words. The BERT model used here is `bert-base-uncased` model, and it uses 12 encoders and no decoders.

C. Literature Review

For this capstone project, a literature review was performed where overall more than 15 articles relevant to Coreference Resolution, attention mechanism, transformer models like BERT, ALBERT were studied

In order to thoroughly understand the Coreference Resolution [1] (anaphora) problem and its background in Natural Language Processing in great depth following articles were reviewed [6], [7], [8], [9], [10], [11].

In addition various articles were reviewed to understand attention mechanism and transformer architectures [12] like BERT [3] and its variants like ALBERT[13] , SpanBERT [14].

Furthermore, an extensive list of articles that were reviewed for this capstone project to understand what the various approaches are to solve coreference resolution [8] and other NLP problems[15]. And how BERT can be fine-tuned [16], [17], [18] and evaluated [19] for such tasks.

D. Goal: Research Questions

The capstone aimed to achieve the following goals:

RQ1 Understand the intricacies of the coreference resolution problem. Moreover, explore various approaches to solve the coreference resolution problem.

RQ1 Explore various state-of-the-art transformer models like BERT and its variants like ALBERT and understand their architecture. Explore how these models can be fine-tuned to perform specific tasks.

RQ2 Adapt BERT and its variant models for coreference resolution tasks and compare the performance of various models against each other.

E. Methodology

This capstone project evaluates transformer models like BERT, its variants like ALBERT, and various implementation of BERT and its variants, including its vanilla model as well as the fine-tuned model and other model implemented by papers from literature review on coreference datasets like GAP coreference dataset among others.

For this capstone project, tools like Jupyter Notebook were used for training & testing models. Python libraries like NumPy, Pandas, Matplotlib were used for data preprocessing & data analysis. Moreover, Python libraries like PyTorch, TensorFlow, scikit-learn, transformers (huggingface), were used for training models.

The aim of this capstone project was to understand and analyze what the coreference resolution problem is, what transformers are, how the coreference resolution problem can be solved using transformers and BERT, and compare & contrast its results with other models/variants of BERT.

II. Data Sets and Data Analysis

A. GAP Dataset

Gender bias is a substantial issue in natural language processing and coreference resolution systems. There appears to be a bias favoring masculine or feminine gender depending on the context and socio-cultural expectations or stereotypes. An example of this can be seen in example 2 and 3 in I-A

The primary dataset used for this project is GAP (Gendered Ambiguous Pronouns) coreference Dataset [20], which is a gender-balanced dataset and is sampled from Wikipedia. Google released this dataset in 2018 To enable a targeted approach to address the gender bias issue and is considered a benchmark for this task. This data provides a text sentence; one ambiguous pronoun from the sentence followed by two candidate names that might or might not refer to the pronoun. An example of a data instance from the GAP dataset can be seen in Table I

TABLE I
Example of a data instance from the GAP dataset.

Text	A	B	Pronoun
Phoebe played **Cheryl Cassidy**, **Pauline**'s friend and also a year 11 pupil in Simon's class. dumped **her** boyfriend ...	Cheryl Cassidy (True)	Pauline (False)	her

This dataset is split into three parts: train, validation, and test dataset, with training and testing sets containing over 2000 records each while the validation set with over 900 records.

This dataset has over 7000 antecedents in the train and test dataset each. Out of these 7000, around 4000 are potential antecedents for the sentences for predicting if they refer to the ambiguous pronoun or not. On performing analysis and exploration on the dataset, some of the following characteristics of the dataset were discovered.

The most frequent name was John; it occurred 140 times in this set.

Following are the other most frequent entities that appear in the (train) dataset:

York : 97 times, William : 91 times, Mary : 86 times
World : 85 times, May : 79 times, August : 74 times
George : 73 times, Paul : 72 times

Some of the least frequent names were: Phoebe, Evanston, Peppy, Trier, Winnetka, Leach, Shoji, Hamada, Kanjiro, Kawai, Angeloz, Justicialist, Menem, Viewers, Humming, Came, Rank, Nuria, Golijov, Ainadamar, among others, that occurred one time.

The graph of entities against their frequencies can be seen in Figure 3. The graph demonstrates that there is a dominance of object pronouns over subject pronouns:

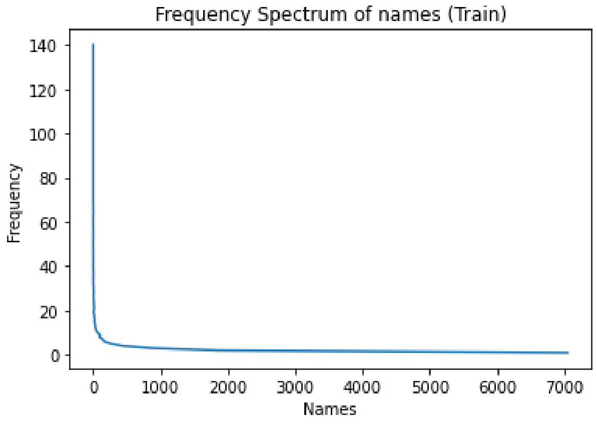

Fig. 3. Frequency Spectrum of names (Training Data Set)

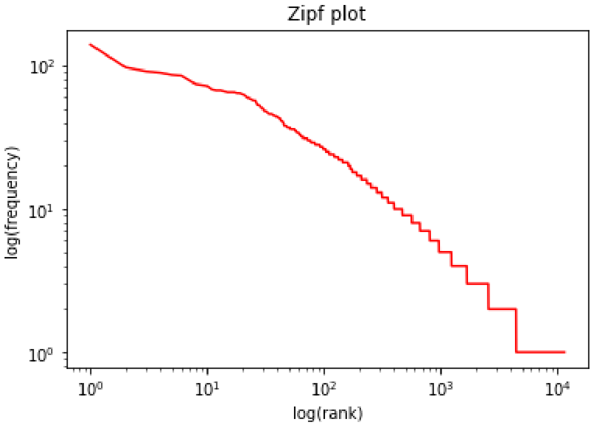

Fig. 4. Zipfian distribution

From viewing the graphs in Figure 4 it appears that the candidate antecedents in this dataset follow Zipf's law which states that, "Across a corpus of natural language, the frequency of any word in that corpus is inversely proportional to its rank in the frequency table. "[21]

The statistical summary of pronouns occurring in text looked can be seen in the Figure II:

TABLE II
DESCRIPTIVE STATISTICS OF PRONOUNS (TRAINING DATA SET)

count	mean	std	min	25%	50%	75%	max
2000.00	3.30	2.11	1.00	2.00	3.00	4.00	17.00

The above table suggests that, on average, there were 3-4 pronouns in every sentence in the dataset, with a minimum of 1 and a maximum of 17 pronouns.

The statistics and counts of pronouns from the ambiguous pronoun column, to which the candidate entities are to be determine look as show in Figure 5:

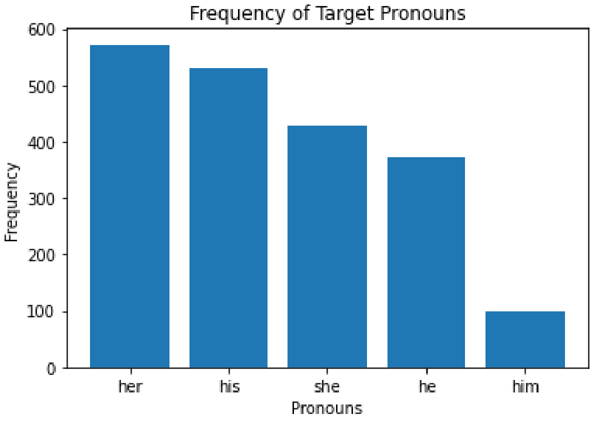

Fig. 5. Frequency of Target Pronouns

Also, since this is a gender-balanced out of the 2000 ambiguous pronouns, there 1000 masculine and feminine pronouns respectively, as shown in the figure 6.

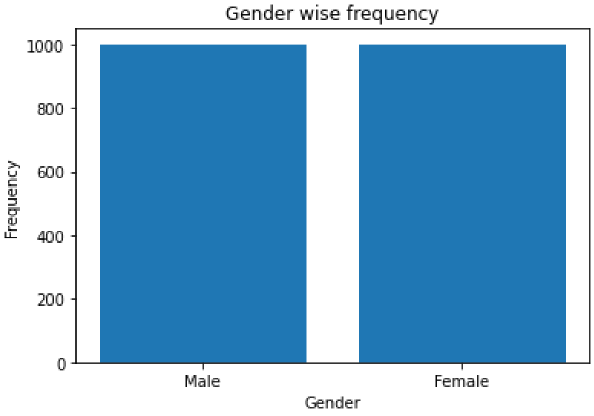

Fig. 6. Gender wise frequency of Pronouns

The models that use this dataset aim to figure whether if the ambiguous pronoun is referring to be antecedent A, antecedent B, or neither of them.

B. ParCorFull[22]

ParCorFull: a Parallel Corpus in English and German language annotated with full coreference. Data is obtained from 3 sources:

- News
- TED Talks
- DiscoMT (Discourse-Oriented Statistical Machine Translation)

This dataset contains parallel texts for the language pair English - German. This language pair is topologically similar to each other, yet they have significant differences in sentence structure, especially when it comes to coreference. Example:

English	German
She wants to, you know, find Obi Wan Kenobi. He's her only hope.	Sie will Obi Wan Kenobi finden. Er ist ihre einzige Hoffnung.

The annotations are performed on this dataset using the annotation tool MMAX2. All annotations for this dataset were performed manually by highly experienced annotators who had a background in linguistic.

III. METHODS

A. Mention-Ranking Coreference Model

Mention-Ranking is considered one of the fundamental algorithms for coreference resolution tasks along with models like mention-pair and entity-mention algorithms. Clark and Manning [8] discussed the mention-ranking coreference algorithm in great depth.

For this project, the `neuralcoref` implementation of this algorithm, from Hugging Face in PyTorch, is used to perform coreference resolution for the GAP coreference task. This model acted as a baseline, and the results of all other vanilla and trained BERT models are compared with the results of this model.

The model follows following steps for finding antecedents referring the mentions:

Step 1 Extract all the potential entities and their pronominal and other (e.g., other entities) references in a given sentence

Step 2 Compute a set of features using Natural Language Processing techniques for each mention pairs.

Step 3 Find the most likely antecedent for each mentioned pair.

Step 4 To the above, I add a final step where I extract the answer to a GAP instance by tracing the coreference chain in the output of `neuralcoref`.

An example of how the `neuralcoref` models performs coreference resolution can be seen below:

1. **Sentence**:
   'My sister has a dog. She loves him. She loves her cat as well. The cat does not like the dog at all.'
   Output spans for each antecedent:
   Antecedent My sister
   Antecedent Span (0, 9)
   mentions_spans [(21, 24), (36, 39), (46, 49)]
   She (21, 24)
   She (36, 39)
   her (46, 49)
   ————————————————
   Antecedent a dog
   Antecedent Span (14, 19)
   mentions_spans [(31, 34), (85, 92)]
   him (31, 34)
   the dog (85, 92)
   ————————————————
   Antecedent her cat
   Antecedent Span (46, 53)
   mentions_spans [(63, 70)]
   The cat (63, 70)
   ————————————————

B. Vanilla BERT Model [3]

For the initial implementation of the model, the `bert-base-uncased` model was used as a tokenizer, and the `BertForMaskedLM` (BERT for Masked Language Model) from the transformers model from the PyTorch library was used as a prediction model. This model performs the coreference resolution task by masking the target reference pronoun with the [MASK] token. After that, the BERT for Masked Language Model is used to predict the entity covered by the [MASK] token. This prediction provides the probabilities of the entities using the softmax function, which adjusts the probability between 0 and 1. The antecedent with higher probability is classified as the antecedent referring to the pronoun.

To improve the performance of this model for the GAP dataset, the concept as suggested in [16] can be used, which replaces the less popular/frequent candidate antecedent names like Rank, Nuria, Golijov, Ainadamar with some popular names like John, Harry, William. These names occur more frequently in corpora like Wikipedia and BookCorupus[5] on which the BERT model is trained and hence allows BERT to predict results more accurately.

C. Vanilla ALBERT (A Lite BERT) Model [13]

The architecture of ALBERT is very similar to that of BERT, except that ALBERT allows parameter sharing. Hence, the input level embedding is lower in dimension (128) when compared to BERT (768). According to [13] this leads to a more than an 80% drop in the number of parameters but a fairly insignificant drop in the performance. The base model of ALBERT has 12M parameters as compared to 108M parameters as used by the base model of BERT. Also, ALBERT allows parameter sharing, which results in to decrease in time for fine-tuning the model.

For performing the coreference resolution task using the model vanilla ALBERT model same steps were performed as for the vanilla BERT model where the pronoun is covered with the [MASK] token and probabilities of antecedents referring to the [MASK] tokens are predicted.

D. BERT / ALBERT + Neural Network

BERT / ALBERT + Neural Network model was helpful to directly compare the performance of BERT with ALBERT. On paper, BERT and ALBERT has very similar architecture, except ALBERT permits sharing of parameters and BERT does not.

TABLE III
BERT V ALBERT ARCHITECTURE

	BERT	ALBERT
Base Model	Layers:12 Hidden:768 Embeddings: 768 Parameter-Sharing:False	Layers:12 Hidden:768 Embeddings:128 Parameter-Sharing:True
Large Model	Layers:24 Hidden:1024 Embeddings: 1024 Parameter-Sharing:False	Layers:24 Hidden:1024 Embeddings:128 Parameter-Sharing:True

This model is very similar to the CorefMulti model. Figure 7 shows architecture of the model. This model, just like the CorefMulti model, performs feature extraction initially and gets all the embeddings for antecedent A, antecedent B, and pronoun in the form of Tensors as well as their labels. On top of that, there is a Multi-Class classifier Neural Network.

The classifier has three layers that are activated by the ReLU function, and batch normalization is performed at every step. The Neural Network uses cross-entropy loss for classification and Adam optimizer.

The Neural Network model is trained with the following hyper parameters:
EPOCHS = 300
BATCH_SIZE = 16
LEARNING_RATE = 0.0007

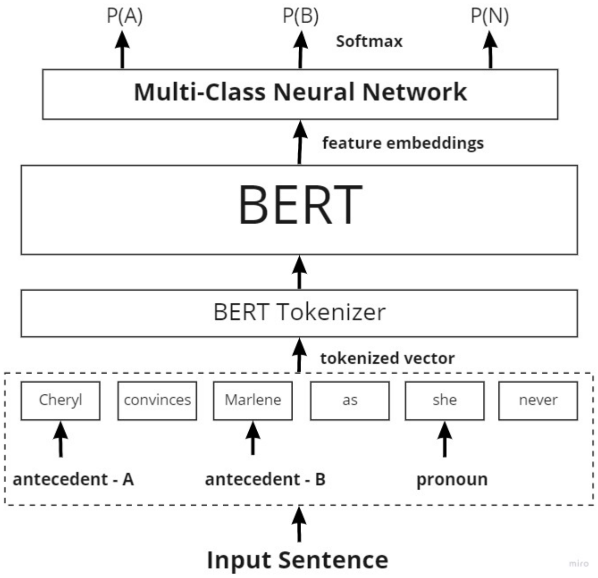

Fig. 7. BERT + Neural Network Architecture: In this architecture word embeddings and features are extracted from BERT models and then a multi-class classifier is applied on top of it to predict the probabilities of each of two potential antecedents referring to the given pronoun, or neither doing so.

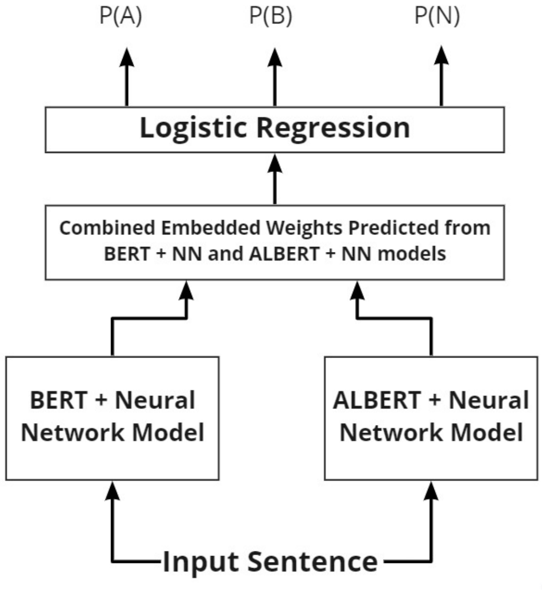
P(A)
P(B)
P(N)
Logistic Regression
Combined Embedded Weights Predicted from BERT + NN and ALBERT + NN models
BERT + Neural Network Model
ALBERT + Neural Network Model
Input Sentence

Fig. 8. Late Fusion Architecture: In this architecture, the outputs from BERT+NN and ALBERT+NN models are captures and concatenated together, and logistic regression is performed on top of that, in order to get benefits of both the models.

E. Late Fusion Model

Late fusion models are based on the principle that many heads are better than one, and hence they are a collaborative attempt to combines the results of different models in order to get benefits from the characteristics of both models. The Late Fusion model for the GAP dataset, as shown in Figure, takes the predicted weights from both BERT + NN and ALBERT + NN models and stores them in a pandas data frame that acts as input (X) for the late fusion model. After that, logistic regression is applied using input (X) from previous steps and the labels (y) are the actual answers from the GAP dataset.

F. CorefMulti BERT Model [23]

To compare the results of baseline models, mention-pair model, and vanilla and trained BERT & ALBERT models, the stage 1 implementation of CorefMulti from [23] was used.

In this model, the sequence features are initially extracted by concatenating token embeddings of the antecedents and pronouns, and the span embeddings are calculated by concatenating the start and end tokens and the results of their element-wise multiplication.

Once these embeddings are extracted, the BERT model is switched to train mode, which starts the fine-tuning phase of BERT by feeding the embedding from the final layer of the BERT encoder to a feed-forward hidden layer (512 units) with ReLU activation. After that, the outputs of the layer are passed through a softmax layer to get probabilities of the antecedents.

This model performs 5-fold cross-validation, which helps the model to understand the data better and reduce any unknown biases.

G. SpanBERT Model [14]

The architecture of the SpanBERT model is similar to that of BERT. The difference is that the SpanBERT model as in [14] masks continuous spans instead of masking random words. The trained and fine-tuned SpanBERT model predicts the entire content of masked span instead of predicting random words. The SpanBERT model is supposed to outperform for various tasks, including coreference resolution.

The following example shows results of the SpanBERT model on ParCorFull Dataset:

2. **Sentence**:
   "Victoria Chen, CFO of Megabucks Banking, saw her pay jump to $2.3 million, as the 38-year-old became the company's president. It is widely known that she came to the Megabucks from rival Lotsabucks.“
   Clusters:

   [((0, 6), 'Victoria Chen , CFO of Megabucks Banking'), ((7, 8), 'her'), ((14, 16), 'the 38 - year - old'), ((25, 26), 'she')] [((4, 6), 'Megabucks Banking'), ((17, 19), 'the company ' s'), ((28, 29), 'Megabucks')]

H. BERT based full coreference model

This model loosely follows the concepts defined in [16] and combines this approach with `BertForMaskedLM` model across all nouns and pronouns phrases for performing full coreference resolution and get coreference clusters on a given text.

Results of BERT based full coreference model on sample sub sentence from GAP dataset:

3. **Sentence**:
   Sally studied with famed family therapist Virginia Satir and began to gain tools for reshaping her life.
   Clusters: virginia satir [('famed family therapist', 3), ('virginia satir', 6)]
   sally [('sally', 0)]

IV. Results

Table IV shows comparison of performance of various BERT models on GAP dataset. All the models were run on a Graphics Processor cluster Tesla P100-PCIE-16GB:

TABLE IV
GAP DATASET EVALUATION

Model	Accuracy	Precision	Recall
Mention-ranking baseline	0.17	0.46	0.37
BERT (base) Vanilla	0.63	0.55	0.63
ALBERT (base) Vanilla	0.59	0.51	0.55
BERT (base) + NN	**0.77**	0.77	**0.76**
ALBERT (base) + NN	0.53	0.53	0.53
Late Fusion Model	0.50	**0.80**	0.48
CorefMulti BERT (base)	0.56	0.54	0.56
CorefMulti BERT (large) [23]	*0.87*	*0.87*	*0.87*

A. Mention-Ranking Coreference Models

The `neuralcoref` implementation model from Hugging Face using PyTorch as described in Clark and Manning [8] 2016, works well on small and simple sentences.

After testing the vanilla `neuralcoref` model on the GAP dataset, the model obtained results with 55% precision and 37% recall.

This model performs well on short sentences, but its performance on large texts, like paragraphs, is not so good; hence it performed poorly on the GAP dataset.

One of the reasons this model does not perform well with long sentences could be that it cannot capture whole information/context for large sequences.

B. Vanilla BERT Model

For this implementaion, the `bert-base-uncased model` was used. Where I have used the `BertTokenizer` and `BertForMaskedLM` implementation of the transformer from PyTorch.

With current model settings, the model gets an accuracy of 63%, precision of 55%, and recall of 63%. This model worked pretty well when one of the two candidate entities was the answer, but being a binary classifier could not handle the cases where neither of the two candidate entities is the correct answer.

Sample Results of Vanilla BERT Model can be seen below:

Entity A: smith
Probability A: 0.7650180459022522
Entity B: rick pitino
Probability B: 0.0009140811744146049
Actual Answer: Smith

Entity A: agathe
Probability: 0.05004461854696274
Entity B: miss dix
Probability: 0.44664663076400757
Actual Answer: Miss Dix

Entity A: big daddy
Probability: 0.0027061912696808577
Entity B: williams
Probability: 0.9731364846229553
Actual Answer: Williams

Entity A: jessica
Probability: 0.0007865129155106843
Entity B: allison
Probability: 0.032582059502601624
Actual Answer: Allison

Entity A: rossetti
Probability: 0.02488189935684204
Entity B: john paul getty ii
Probability: 0.0066771190613508224
Actual Answer: Neither

As discussed in III-B the performance of this model can be improved by replacing the less frequent entities with more common first names and last names, which is suggested in the paper [16] because the BERT is trained with Wikipedia and BookCorpus dataset.

Also, performing a multi-class classification would provide a better performance in cases where neither of the two entities is the answer.

C. Vanilla ALBERT (A Lite BERT) Model [13]

For this model, I have used the `albert-base-v2 model`. Where I have used the `AlbertTokenizer` and `AlbertForMaskedLM` implementation of the transformer from PyTorch.

With current model settings, the model gets an accuracy of 59%, precision of 51%, and recall of 55%. The ALBERT model performs slightly worse than the BERT base model.

D. CorefMulti BERT Model [23]

The CorefMulti BERT Model [23] uses the stage 1 implementation of CorefMulti from [23]. The CorefMulti BERT Model uses a feed-forward hidden layer (512 units) with ReLU activation over the BERT model and trains the model with the following hyper parameters: learning_rate = 1e-5, batch size 4. This model also performs 5 fold cross validation for bias reduction.

The CorefMulti BERT model shows a significantly better performance than any other baseline model. This model shows over 21% improvement over the baseline models of BERT and ALBERT.

TABLE V
EXAPMLE OUTPUTS OF BERT (BASE) + NN ON THE GAP DATASET.

Text	A	B	N
... Olympic-medalist Bob Suter are Dehner's uncles. His cousin is ...	Bob Suter P(A)=.84 Incorrect	Dehner P(B)=.16 Correct	Neither P(N)≈0 Incorrect
... Grenfell's career as a monologuist was directly inspired by Draper. Her nephew ...	Grenfell P(A)=.99 Correct	Draper P(B)≈0 Incorrect	Neither P(N)≈0 Incorrect
... Swedish divas Robyn and Lykke Li. Perry did a great job of letting us know she's ...	Robyn P(A)=.00 Incorrect	Lykke P(B)≈.1 Incorrect	Neither P(N)≈.99 Correct

E. BERT / ALBERT + Neural Network

Table V demonstrates example outputs of the BERT(base) + NN model on sample rows. The model predicted correctly for Example 1, incorrectly for Example 2, and one sample where neither of antecedent A and antecedent B refers to the pronoun.

Table VI compares the performance of BERT + NN and ALBERT + NN model, in terms of their performance and time required to train the model on Graphical Processor cluster Tesla P100-PCIE-16GB.

TABLE VI
BERT V ALBERT RESULTS

	BERT	ALBERT
Model	bert-base-uncased	albert-base-v2
Time to Train	15 minutes	10 minutes
Accuracy	0.77	0.53
Precision	0.77	0.53
Recall	0.76	0.53

After performing the literature review and using the trained model for BERT and ALBERT, it was observed that even though ALBERT is called as A Lite BERT, it only decreases the embedding by allowing it to share. However, its main objective is to scale the model and has the same number of hidden layers and thus take a roughly similar time to train the model as BERT but provide worse results.

Also, it should be noted that after fine-tuning the bert-base model, the BERT + NN model performs way better than the CorefMulti on the bert-base model.

V. DELIVERABLES

The deliverable includes datasets, code for fine-tuned transfer models, project report with results, analysis, and comparison of different datasets & transformer models.

VI. CONCLUSION

After understanding the intricacies of the Coreference Resolution problem and understanding the architecture and working of state-of-the-art transformer models like BERT and its variants like ALBERT, it can be concluded that BERT, when fine-tuned, can be used effectively for coreference resolution tasks (like GAP coreference) quite well and provide above baseline results.

On analyzing the performance of BERT variant ALBERT, it was observed that. While called (A Lite BERT), ALBERT has the same number of hidden layers but allows parameter sharing to reduce the model size and the number of parameters. ALBERT models take less time to fine-tune while using the base model of ALBERT as compared to the base model of BERT. The performance of fine-tuned ALBERT model worse than the performance of the BERT model.

VII. FUTURE WORK

Retrained BERT models like SpanBERT can effectively extract the coreference clusters from sentences, but further work needs to be done to perform full coreference and make the SpanBERT model work for non-English languages German.

Gender bias is one of the main problems in Natural Language Processing systems. Still, there are other biases like racial bias, cultural bias, socio-economical bias, etc., and exploration needs to be done to find out how these biases affect the performance of Natural Language Processing models and how they can it affect coreference resolution systems.

One of the approaches could be to generate synthetic data of additional instances to negate such inherent biases to tackle these biases. Trained and fine-tuned BERT models could analyze the synthetically modified custom test data set.

Transformer models like BERT are very good at creating interdependent embeddings from texts and understanding text. These models can be used to more remarkable effect for various problems like semantic analysis, emotion detection, and recognition, sarcasm identification.

Alternatively, exploration can be done to analyze how BERT can be used for another sequence task, such as temporal event analysis.

REFERENCES

[1] P. Elango, "Coreference resolution : A survey," 2006.

[2] H. J. Levesque, E. Davis, and L. Morgenstern, "The winograd schema challenge," in *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, ser. KR'12. AAAI Press, 2012, p. 552–561.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[4] S. Liu, N. Yang, M. Li, and M. Zhou, "A recursive recurrent neural network for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1491–1500. [Online]. Available: https://www.aclweb.org/anthology/P14-1140

[5] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[6] R. Zhang, C. N. dos Santos, M. Yasunaga, B. Xiang, and D. Radev, "Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering," 2018.

[7] W. Wu, F. Wang, A. Yuan, F. Wu, and J. Li, "CorefQA: Coreference resolution as query-based span prediction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 6953–6963. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.622

[8] K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2256–2262. [Online]. Available: https://www.aclweb.org/anthology/D16-1245

[9] H.-L. Trieu, A.-K. Duong Nguyen, N. Nguyen, M. Miwa, H. Takamura, and S. Ananiadou, "Coreference resolution in full text articles with BERT and syntax-based mention filtering," in *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 196–205. [Online]. Available: https://www.aclweb.org/anthology/D19-5727

[10] H. Chen, Z. Fan, H. Lu, A. L. Yuille, and S. Rong, "Preco: A large-scale dataset in preschool vocabulary for coreference resolution," 2018.

[11] Y. T. Cao and H. Daumé III, "Toward gender-inclusive coreference resolution," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 4568–4595. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.418

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[13] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2020.

[14] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.tacl-1.5

[15] T. Klein and M. Nabi, "Attention is (not) all you need for commonsense reasoning," 2019.

[16] F. Alfaro, M. R. Costa-jussà, and J. A. R. Fonollosa, "BERT masked language modeling for co-reference resolution," in *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 76–81. [Online]. Available: https://www.aclweb.org/anthology/W19-3811

[17] Y. Arase and J. Tsujii, "Transfer fine-tuning: A BERT case study," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5393–5404. [Online]. Available: https://www.aclweb.org/anthology/D19-1542

[18] T. I. Denk and A. Peleteiro Ramallo, "Contextual BERT: Conditioning the language model using a global state," in *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*. Barcelona, Spain (Online): Association for Computational Linguistics, Dec. 2020, pp. 46–50. [Online]. Available: https://www.aclweb.org/anthology/2020.textgraphs-1.5

[19] N. Jiang and M.-C. de Marneffe, "Evaluating BERT for natural language inference: A case study on the CommitmentBank," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6086–6091. [Online]. Available: https://www.aclweb.org/anthology/D19-1630

[20] K. Webster, M. Recasens, V. Axelrod, and J. Baldridge, "Mind the gap: A balanced corpus of gendered ambiguous pronouns," 2018.

[21] G. K. Zipf, *The Psychobiology of Language*. New York, NY, USA: Houghton-Mifflin, 1935.

[22] E. Lapshinova-Koltunski, C. España-Bonet, and J. van Genabith, "Analysing coreference in transformer outputs," 2019.

[23] R. Chada, "Gendered pronoun resolution using BERT and an extractive question answering formulation," in *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 126–133. [Online]. Available: https://www.aclweb.org/anthology/W19-3819