Slide 1: Title
Welcome to our project summary on Statistical Natural Language Processing, where we would focus on evaluating general language understanding through various BERT model adaptations.

Slide 2: Introduction
BERT has become a trend since its introduction by Google AI Language in 2018. The core idea of BERT is its bidirectional training, which considers both left and right context simultaneously using the Masked language modelling and next sentence prediction. Moreover, BERT's architecture is built solely on transformer encoder. To process input, BERT incorporates token, segment, and positional embeddings, which help distinguish between different sentences and their order in the text.

Slide 3: Project goal
The project goal is to fine-tune BERT for sentence classification based on GLUE benchmark. To achieve this, a simple linear neural network layer is added on top of the BERT model. This layer uses the output from the [CLS] token and map to 2 classes for binary classification. Finally, class prediction is made by selecting the class with the highest softmax probability

Slide 4: Two datasets

In this slide, we're looking at two datasets: the Quora Question Pair Dataset (QQP) and the Stanford Sentiment Treebank binary (SST-2). For the QQP task, we aim to determine whether pairs of questions are duplicates. On the other hand, the SST-2 task involves sentiment analysis of movie reviews, where the model predicts if a review is positive or not

Slide 5: BERT and its variants

Another purpose of our project is to also benchmark these five models on the two datasets to find out which model performs the best. As a result, we now explore BERT and its variants

Firstly, DeBERTa distinguishes itself with a disentangled attention mechanism and RoBERTa is optimized for more robust performance. DistilBERT streamlines BERT by offering a distilled version that retains 97% of its performance and lastly, ALBERT refines BERT by sharing parameters across layers, significantly reduced to only 11 million parameters.

Slide 6: Experiment setup 1
In the experiment setup, we would first need to choose the BERT variant model. Our code uses the two generic imports, AutoTokenizer and AutoModelForSequenceClassification. By structuring our code like this, it is extremely easy to switch among the BERT models without rewriting the entire code.

Slide 7: Experiment setup 2
Since BERT models work with tokenization, this stage aims to tokenize sentences for further processing. These tokenizers can add special tokens necessary for BERT, and also add padding and truncating sequences to a fixed length.

Slide 8: Experiment setup 3
Finally we set up the optimizer and learning rate scheduler. Then, we define the training loop, which includes batch training with gradient calculation and model updates. Then, we perform validation to evaluate the model on unseen data. After training all BERT variants on the two datasets, we save their results for analysis

Slide 9: Results from QQP
After training, We can conclude that the base BERT model and DeBERTa has the highest training and inference time, while RoBERTa, DistilBERT and ALBERT have only about half training and inference time. Taking into account both accuracy and computation efficiency, we believe DeBERTa and RoBERTa perform the best on QQP dataset

Slide 10: Results from SST-2
However, in SST-2, it is quite hard to tell which model performs the best judging solely by testing metrics. Therefore, we can rely on the training and inference time as well. We finally conclude that DistilBERT and RoBERTa performs best on SST-2 dataset.

Slide 11: Discussions
In conclusion, we observed that RoBERTa and DeBERTa consistently delivered robust performances on both the QQP and SST-2 tasks. While DeBERTa had slightly higher accuracy and F1 scores, smaller models like DistilBERT and ALBERT also achieved competitive results. Our findings confirm the strong generalization ability of all BERT variants across a range of NLP tasks, which is important for real-world applications.

Slide 12: Available tutorial
For curious future readers, we have already prepared another tutorial slide on how to replicate our workflow and fine tune the models on Kaggle. It is hosted on Github from the appendix section.

Slide 13: Thank you slide
That is everything about our project on evaluating and benchmarking BERT models. Thank you so much for your listening.