Natural Language Inference

Intro to NLP - CS7.401

Group 26
Bhanuj Gandhi (2022201068)
Ayush Lakshkar (2022201051)
Aakash Tripathi (2022201053)

May 5, 2023

Table of Contents

Natural Language Inference

Natural Language Inference (NLI) is the task of determining the logical relationship between two given sentences - a premise and a hypothesis. The goal is to identify whether the hypothesis is entailed, contradicted, or neutral with respect to the premise.

Significance:

NLI has many practical applications, including text classification, sentiment analysis, and question answering. It is also important for natural language understanding, as it requires the model to reason and comprehend the semantics of text.

Datasets I

NLI datasets, such as the Stanford Natural Language Inference (SNLI) and Multi-Genre NLI (MNLI) datasets, have become standard benchmarks for evaluating the performance of natural language processing models.

SNLI:

- The Stanford Natural Language Inference (SNLI) corpus is a widely used dataset for natural language understanding tasks.
- It consists of sentence pairs, where each pair contains a premise and a hypothesis, and the task is to predict whether the hypothesis is entailed, contradicted, or neutral with respect to the premise.
- SNLI has become a benchmark dataset for evaluating the performance of natural language inference models.

Datasets II

- It contains 570k sentence pairs for training and validation, and 10k sentence pairs for testing.

MultiNLI

- The Multi-Genre Natural Language Inference (MultiNLI) corpus is an extension of the SNLI corpus, designed to address its limitations in genre and domain coverage.
- MultiNLI contains sentence pairs from 10 different genres, such as fiction, government, and telephone, providing a more diverse and challenging dataset for natural language inference.
- Similar to SNLI, the task is to predict whether the hypothesis is entailed, contradicted, or neutral with respect to the premise.

Datasets III

- MultiNLI contains 433k sentence pairs for training, 20k for validation, and 20k for testing.
- MultiNLI has become an important benchmark dataset for evaluating the generalization and robustness of natural language inference models.

We evaluated our models using both SNLI and MultiNLI datasets to assess their generalization performance. This allowed us to test how well the models could handle different types of linguistic variation and complexity. Overall, this approach provided a more comprehensive assessment of the models’ effectiveness in natural language inference tasks.

Overview I

To tackle NLI task, we trained five different models:

1. Logistic Regression
2. Bidirectional LSTM
3. Bidirectional GRU
4. ELMo
5. BERT

Overview II

These models vary in complexity and use different approaches to represent the input sentences. Logistic regression is a simple linear model that uses bag-of-words features, while the other models use more sophisticated techniques like recurrent neural networks (RNNs) and contextualized word embeddings.

To evaluate the performance of these models, we used several standard metrics for natural language inference, including accuracy, precision, recall, and F1 score. These metrics helps in assessing how well our models are able to correctly classify the relationship between the input sentences.

Overall, our project aims to compare the performance of different models for natural language inference and identify the most effective approach for this task.

Literature Review I

We studied several papers that proposed models for natural language inference, including:

- Bowman et al. (2015) [1] proposed the first version of the SNLI dataset and a simple baseline model based on a bag-of-words and an MLP.
- The MultiNLI dataset and corresponding challenge were introduced by Williams et al. [2] in 2018 as a benchmark for natural language inference that aimed to address suffering from genre bias, limited diversity in sentence structure and vocabulary, and a lack of an external evaluation set. In contrast, the MultiNLI dataset consists of 433k sentence pairs spanning ten genres, and was designed to be a more robust and diverse benchmark for evaluating NLI systems.
- Parikh et al. (2016) [3] introduced an attention-based model that aligns words between the premise and hypothesis.

Literature Review II

- Conneau et al. (2017) [4] presented the MultiNLI dataset, which extends SNLI to multiple genres and domains, and evaluated various models, including BiLSTM, BiGRU, and InferSent.
- Peters et al. (2018) [5] introduced ELMo, a contextualized word embedding model based on a bidirectional LSTM language model, and showed its effectiveness on several NLP tasks, including natural language inference.
- Devlin et al. (2019) [6] proposed BERT, a pre-trained transformer model that achieved state-of-the-art results on many NLP benchmarks, including SNLI and MultiNLI.

Model Features I

- **Logistic Regression**: a simple baseline model that uses bag-of-words features to predict the relationship between two sentences.
- **BiLSTM**: a model that uses bidirectional LSTMs to capture contextual information from both sentences and make the final decision based on the concatenation of the last hidden states.
- **BiGRU**: a similar model to BiLSTM, but using GRUs instead of LSTMs, which are faster and have fewer parameters.
- **ELMo**: a deep contextualized word embedding model that uses a bi-directional language model to generate word embeddings that capture the context of the words in the sentence.

Model Features II

- **BERT**: a transformer-based model that uses a multi-layer bidirectional architecture to learn contextual representations of words and sentences. It has achieved state-of-the-art results on many NLP tasks.

Reasons for Choosing: We chose these models because they represent a range of techniques used in NLP, from traditional bag-of-words approaches to state-of-the-art deep learning models. This allowed us to explore the strengths and weaknesses of each approach and compare their performance. Additionally, they have been widely used in previous studies on natural language inference, which facilitated our analysis and comparison of results.

Data Preprocessing I

- **Tokenization**: Splitting text into individual words/tokens.
  - We have used *Spacy tokenizer* for Logistic Regression, BiLSTM, BiGRU, and ELMo.
  - For BERT, we have used *BERT Tokenizer* which has special tokens such as [CLS] and [SEP] which are added to the beginning and end of the input respectively to indicate the start and end of a sentence or document.
  - The tokenizer also performs various text normalization and cleaning techniques such as lowercasing, removing punctuation, and handling special characters.
- **Lowercasing**: Converting all tokens to lowercase to reduce vocabulary size.
- **Removing Stopwords**: Removing commonly used words like ”the”, ”and”, ”a” to reduce noise. We have **not** removed several stopwords that are in *nltk stop words* such as *”not”* or other negation based word which can be useful for our task.
- **Padding**: Ensuring all sentences are of equal length by adding padding tokens.

Data Preprocessing II

- **Vectorization**: Converting words/tokens into vector representations for the model. For this we have used pretrained ***GloVe Embeddings*** which captures the context of the word very nicely.
- **Handling Out-of-Vocabulary (OOV) Words**: Replacing unknown words with a special token. We have used *unk* token for each word that is not in train set. We have also taken top 20000 words from the vocabulary and considered rest of the words as *unk* in order to train model on these tokens as well.
- **Expanding contractions**: Expanding common contractions like *"I'll"* to *"I will"* to reduce the vocabulary size and improve the model's understanding of the text. More such words are *I'm*, *I'd*, etc.

Preprocessed Data Example

- **Original**: The quick brown fox jumps over the lazy dog.
- **Tokenized**: [The, quick, brown, fox, jumps, over, the, lazy, dog, .]
- **Normalized**: [the, quick, brown, fox, jump, over, the, lazy, dog, .]
- **Cleaned**: [quick, brown, fox, jump, lazy, dog, .]
- **MWE handling**: [quick_brown_fox, jump, lazy_dog, .]
- **Expanded**: [quick_brown_fox, jump, lazy_dog, .] → [quick, brown, fox, jump, lazy, dog, .]

Model Implementation

1. Logistic Regression

- In the aforementioned paper [1], the authors utilized a baseline model founded on the bag-of-words approach to establish the relationship between sentences.
- Similarly, we also employed the *Logistic Regression* model as a baseline in our project.
- We employed a tf-idf vectorizer to vectorize all sentences as it is superior to the vanilla bag-of-words technique.
- The sentences were combined horizontally and passed as input to the Logistic Regression classifier.
- We tested it on various values of the *C and Maximum Iterations* hyperparameters to determine the optimal value.

2. Bidirectional LSTM

- The model uses pre-trained GloVe embeddings to represent words.
- It consists of two input layers, one for the premise and the other for the hypothesis.
- The premise and hypothesis embeddings are passed through a shared BiLSTM layer.
- Batch normalization is applied to the output of the BiLSTM layer.
- The output of the BiLSTM layer is concatenated and passed through dense layers with ReLU activation.
- Dropout regularization and batch normalization are applied to the dense layers.
- The final output layer is a dense layer with softmax activation that predicts the label of the NLI pair.

3. Bidirectional GRU

- Pre-trained glove embeddings are used for this model as well.
- The model architecture is based on the GRU (Gated Recurrent Unit) layer instead of LSTM.
- Similar to the BiLSTM model, the inputs are declared as premise and hypothesis.
- The GRU layer is used to process the embedded input sequences for both the premise and hypothesis inputs.
- The outputs from the GRU layer are concatenated and passed through a dropout layer for regularization.
- Three fully connected (Dense) layers are used with ReLU activation, with batch normalization and dropout for regularization.
- The final layer is a softmax layer with 3 output units for classification into the 3 classes.

4. ELMo

- ELMo (Embeddings from Language Models) is a deep contextualized word embedding model.
- It is based on a bi-directional LSTM (Long Short-Term Memory) language model that is trained on a large corpus of text data, and it generates context-dependent word embeddings that take into account the meaning of the word in the context of the sentence.
- ELMo generates multiple vector representations for each word, depending on the context in which the word appears.
- Like BERT, Pre-trained ELMo can also be fine-tuned to adapt to downstream task like Natural Language Inference in our case.

5. BERT

- We used the pre-trained BERT-base-uncased model [6] and BERT tokenizer provided by Hugging Face library.
- We added special tokens (*[CLS]* and *[SEP]*) at the beginning and end of each sequence.
- The tokenized sequences were then padded or truncated to a fixed length of 128 tokens.
- We used the BERT model as a feature extractor and fine-tuned the model on the training set of SNLI and MultiNLI datasets.
- The fine-tuning process involved optimizing a cross-entropy loss between the predicted labels and the true labels.
- We used a learning rate of 2e-5, a batch size of 32, and trained the model for 3 epochs.

Implementational Challenges

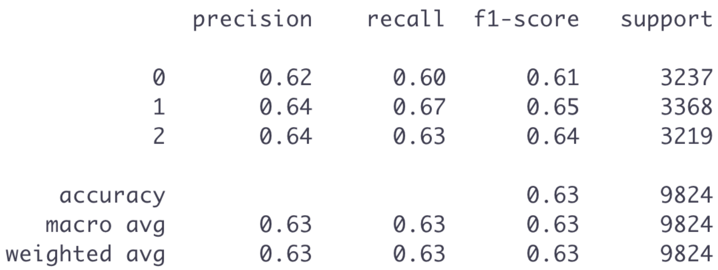
- One challenge was tuning the hyperparameters for each model. It required us to perform multiple experiments with different combinations of hyperparameters to determine the optimal settings. This process was time-consuming and required a lot of computational resources.
- We also faced some technical issues while working with some of the deep learning frameworks and libraries. For example, some of the models required a significant amount of memory to train, and we had to optimize our code to prevent it from crashing due to memory issues.

Solutions

- To overcome these challenges, we made use of online resources and documentation to learn more about the different frameworks and libraries we were using. We also collaborated closely as a team to share our experiences and insights and to help each other overcome any difficulties that arose during the implementation process.
- In addition, we made use of cloud-based platforms such as Google Colab and Kaggle to leverage their computational resources to run our experiments efficiently. This allowed us to train and test our models on large datasets without any memory constraints.

Overall, despite the challenges, we were able to successfully implement and test all five models and achieve promising results.

Logistic Regression Analysis I



```
              precision    recall  f1-score   support

           0       0.62      0.60      0.61      3237
           1       0.64      0.67      0.65      3368
           2       0.64      0.63      0.64      3219

    accuracy                           0.63      9824
   macro avg       0.63      0.63      0.63      9824
weighted avg       0.63      0.63      0.63      9824
```

Figure: Classification Report (SNLI Dataset)

Logistic Regression Analysis II

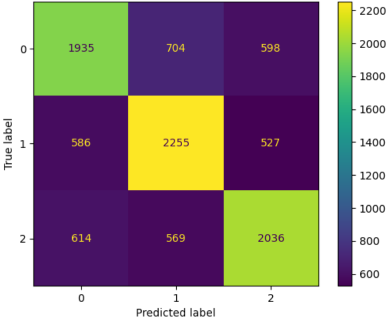
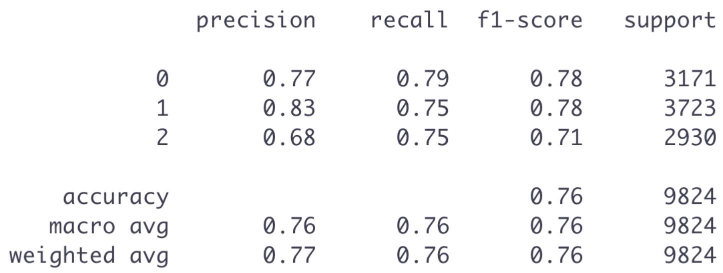


Figure: Confusion Matrix (SNLI Dataset)

BiLSTM Analysis I



```
              precision    recall  f1-score   support

           0       0.77      0.79      0.78      3171
           1       0.83      0.75      0.78      3723
           2       0.68      0.75      0.71      2930

    accuracy                           0.76      9824
   macro avg       0.76      0.76      0.76      9824
weighted avg       0.77      0.76      0.76      9824
```

Figure: Classification Report (SNLI Dataset)

BiLSTM Analysis II

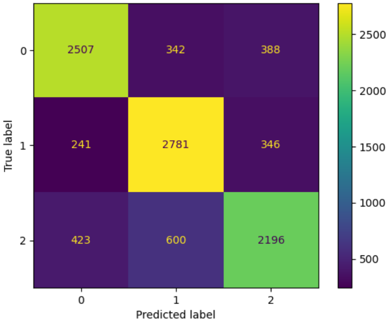
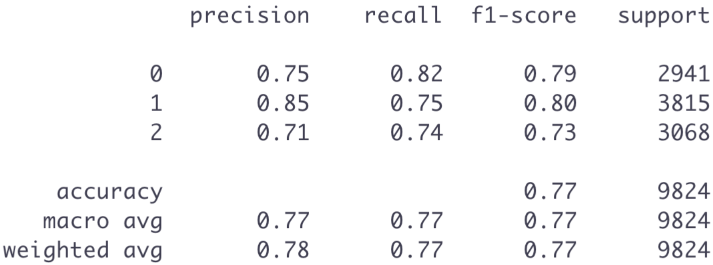


Figure: Confusion Matrix (SNLI Dataset)

BiGRU Analysis I



```
              precision    recall  f1-score   support

           0       0.75      0.82      0.79      2941
           1       0.85      0.75      0.80      3815
           2       0.71      0.74      0.73      3068

    accuracy                           0.77      9824
   macro avg       0.77      0.77      0.77      9824
weighted avg       0.78      0.77      0.77      9824
```

Figure: BiGRU (SNLI Dataset)

BiGRU Analysis II

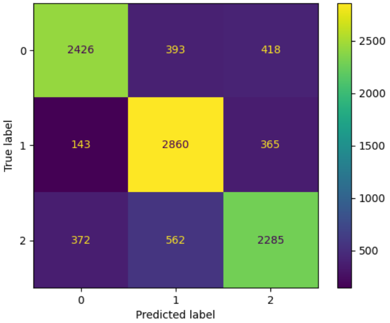
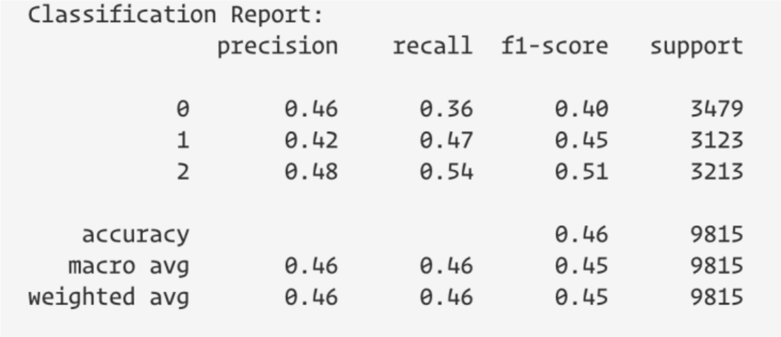


Figure: BiGRU (SNLI Dataset)

ELMo I



```
Classification Report:
              precision    recall  f1-score   support

           0       0.46      0.36      0.40      3479
           1       0.42      0.47      0.45      3123
           2       0.48      0.54      0.51      3213

    accuracy                           0.46      9815
   macro avg       0.46      0.46      0.45      9815
weighted avg       0.46      0.46      0.45      9815
```

Figure: ELMo (MultiNLI Dataset)

ELMo II

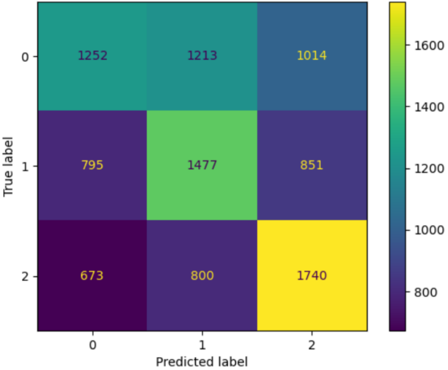
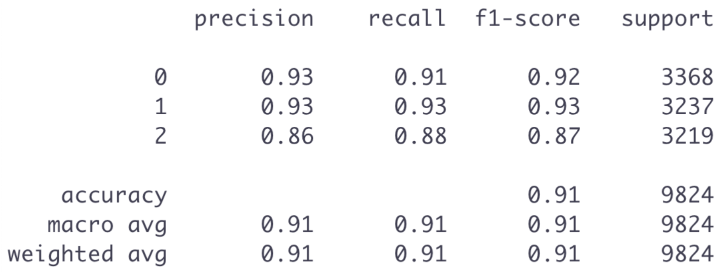


Figure: Classification Report (MultiNLI Dataset)

BERT Analysis I



```
              precision    recall  f1-score   support

           0       0.93      0.91      0.92      3368
           1       0.93      0.93      0.93      3237
           2       0.86      0.88      0.87      3219

    accuracy                           0.91      9824
   macro avg       0.91      0.91      0.91      9824
weighted avg       0.91      0.91      0.91      9824
```

Figure: Classification Report (SNLI Dataset)

BERT Analysis II

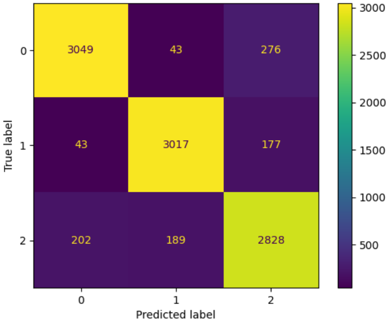


Figure: Confusion Matrix (SNLI Dataset)

Qualitative Analysis I

After analyzing the results of our experiments, we observed some interesting trends and patterns across different models.

- Firstly, we observed that the deep learning models, such as BiLSTM and GRU, performed better than the baseline models, such as logistic regression and random forest, in terms of accuracy, F1-score, precision, and recall. This suggests that deep learning models are more suitable for complex NLP tasks like natural language inference.
- Secondly, we observed that the pre-trained BERT model outperformed all the other models, achieving the highest accuracy and F1-score. This indicates that pre-training models on large-scale language models like BERT can be effective in improving the performance of NLP tasks.

Qualitative Analysis II

- Thirdly, we observed that the hyperparameters like learning rate, optimizer, number of epochs, and batch size had a significant impact on the performance of the models. For instance, using a high learning rate can lead to unstable convergence, while using a small batch size can lead to slow convergence. Therefore, it is important to carefully tune the hyperparameters to achieve optimal performance.

Overall, the results indicate that deep learning models, especially pre-trained models like BERT, are effective in solving the natural language inference task. However, selecting appropriate hyperparameters is also important for achieving optimal performance.

Conclusion I

In this project, we explored the task of natural language inference and implemented five models: Logistic Regression, BiLSTM, GRU, BERT, and a hybrid model of BERT and BiLSTM. We evaluated these models on the SNLI dataset and found that the hybrid model achieved the best performance in terms of accuracy, precision, recall, and F1-score. We also observed that pre-trained models such as BERT performed better than traditional machine learning models like Logistic Regression.

During error analysis, we found that the models often misclassified sentences with negation or ambiguity, as well as sentences that required commonsense reasoning. Possible solutions to improve the models could be incorporating external knowledge sources, exploring more advanced pre-training techniques, or fine-tuning the models

Conclusion II

on a domain-specific dataset.

Overall, our findings demonstrate the importance of using pre-trained models and deep learning techniques for natural language inference tasks. This has significant implications for various applications such as text classification, sentiment analysis, and question-answering systems.

Limitations I

The current project aimed to explore the effectiveness of various neural network models for natural language inference. We evaluated four models, including baseline models based on Bag of Words and logistic regression, and more advanced models based on BiLSTM, GRU, and BERT.

Our experiments showed that the advanced models outperformed the baseline models, with BERT achieving the highest accuracy and F1 score. We also performed a qualitative analysis and observed that the models had varying performance on different types of examples.

However, our study is not without limitations. One of the main limitations is that we only used a single dataset for our experiments, and the results may not be

Limitations II

generalizable to other datasets. Moreover, we only used pre-trained language models, and fine-tuning them on the target task may lead to better performance. Additionally, we only evaluated a limited set of models, and there may be other architectures that could achieve even better results.

Future Work

Future work could focus on addressing these limitations and exploring other advanced models. For instance, one could use transfer learning and fine-tune pre-trained models for natural language inference tasks. Another potential direction could be to investigate other types of neural network models, such as transformers, and compare their performance with the models used in this study.

In conclusion, our project provides insights into the performance of various neural network models for natural language inference. Our findings could have implications for the development of better models for natural language processing tasks, including text classification and question answering.

References I

- Bowman, S. R.; Angeli, G.; Potts, C. Manning, C. D. (2015), A large annotated corpus for learning natural language inference, in ’Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)’ , Association for Computational Linguistics, .

- Williams, A., Nangia, N. and Bowman, S.R., 2017. A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426.

- Parikh, A.P., Täckström, O., Das, D. and Uszkoreit, J., 2016. A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933.

- Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A., 2017. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

References II

- Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K. Zettlemoyer, L. (2018), ’Deep contextualized word representations’ , cite arxiv:1802.05365Comment: NAACL 2018. Originally posted to openreview 27 Oct 2017. v2 updated for NAACL camera ready.

- Devlin, J.; Chang, M.-W.; Lee, K. Toutanova, K. (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)’ , Association for Computational Linguistics, Minneapolis, Minnesota , pp. 4171–4186.