Project Milestone 3
Team: Hex HyperCity

Mason Lien
Matthew Penne
Mrinal Rawool
Ketemwabi Yves Shamavu

Contents

Abstract

Natural language inference (NLI) is the process of inferring the meaning of a sentence. Transfer learning consists of importing a model pretrained on a similar task and modifying it to the desired dataset. Bidirectional encoder representations from transformations (BERT) is a large model developed by Google for encoding sentences. This project uses transfer learning from various BERT models and additional output layers for NLI. In the NLI task, two statements, a hypothesis and a premise, are fed to the model. The model subsequently predicts whether the hypothesis is true, false, or undetermined given the premise. When the hypothesis is true, it is said to be entailed by the premise. When it is false, it is contradicted by the premise, and when it is undetermined, it is neutral with respect to the premise.

Chapter 1

Milestone 1: Project Ideas

1.1 Introduction

Project ideas were inspired from online datasets sharing platforms such as Kaggle and group members personal research. The group also looked at published literature for other curated datasets, including the papers by [28, 10, 27, 22, 17].

Since the data sets mentioned by [22] and [17] required extensive module-specific CITI training and given the limited timeframe for our project, we excluded these papers from the project ideas detailed below.

1.2 Project Idea 1: Natural Language Inference using Deep Learning

Introduction

Natural Language Processing (NLP) is a domain that deals with the field of computer science and linguistics to devise ways for humans to interact with machines using human language. NLP includes some low-level tasks whose objective is to learn linguistic context. Examples of such tasks are parts-of-speech tagging, named entity recognition, information extraction, relationship extraction etc. Such tasks fall under shallow learning as the machine does not seek to understand the meaning or context of the content. On the other hand, high-level tasks such as question-answering, reading comprehension, and inference require the use of reasoning and knowledge in order to achieve the intended objective. According to Oxford dictionary, inference is defined as the process of reaching a conclusion based on evidence and reasoning. Currently, research within the realm of NLI can be broadly classified in two categories; textual inference and plausible inference[23]. Textual inference tasks is characterised by a definite, concrete hypothesis for every premise-hypothesis pair while plausible inference tasks require abductive reasoning and external knowledge. In this project, our objective is to build a deep learning model to solve a 3-way textual inference

task.

Problem Statement
The inference task can be described as semantically determining whether a **hypothesis** statement is **true**, **false**, or **undetermined** given the **premise**. A true, false, and an undetermined statement is termed as *e*ntailment, contradiction, and **neutral**, respectively.

Examples:
*P*remise: A man inspects the uniform of a figure in some East Asian country
*H*ypothesis: The man is sleeping
*L*abel: Contradiction
*P*remise: A soccer game with multiple males playing.
*H*ypothesis: Some men are playing a sport.
*L*abel: Entailment
*P*remise: A black race car starts up in front of a crowd of people.
*H*ypothesis: A man is driving down a lonely road.
*L*abel: Neutral

Applications
Natural Language Inference tasks are a stepping stone towards progress in question answering or semantic search tasks. Any task that requires the model to look beyond keywords and syntax and grasp the meaning and context of text uses some kind of inference-based technique to solve the task at hand. NLI is also used as an evaluation technique for machine translation task.

Approaches and Data set
The team plans to use a benchmark data set released in 2015 called Stanford Natural Language Inference corpus, also known as SNLI [11]. This data set is a collection of 570k human-written English sentence pairs manually labeled as entailment, contradiction, and neutral. The corpus includes a train, test, and development split with the last two having close to 10K examples each [6]. Additionally, we would be able to track our model performance by comparing it with the performance metrics of other models that use SNLI. This information is made available through a leaderboard [4].

1.3 Project Idea 2: Evaluating Freeze Injury in Winter Wheat with Deep Learning

Introduction
Freeze injury (FI) is an abiotic stress that significantly impacts normal growth in plants [15]. For winter wheat grown in the Midwest, the probability of experiencing FI is high, since it is planted in the fall where normal growth consists

of germination, emergence, and tillering [9]. Low temperatures are known to kill plants by damaging the crown of the wheat plant, which is the main point of growth. The severity of the injury is influenced by both low temperatures and the duration of low temperatures. When FI occurs, stem growth can be delayed or terminated, resulting in plants tillering at different times. This response causes moderate to severe yield reductions and prolongs harvest due to fields having a non-uniform dry down [9], reducing profits and food supplies.

In regard to the spring of 2020, most of the state of Nebraska experienced significant freezing temperatures between April 10-16th, where lows were below 20 degrees Fahrenheit for an extensive period of time. At the University of Nebraska-Lincoln Havelock research farm, current growth stages were at a tillering stage for the winter research plots being conducted for agriculture studies. At this growth stage, the research plots were moderatly susceptible to FI and had increased the risk of significant injury the closer to jointing. Typical injury observed during this growth stage is leaf burning, where leaf tissue turns from a healthy green to yellow and finally brown if the plant does not recover. One of the main objectives of the UNL small-grains breeding program is to develop wheat varieties that are tolerant to these types of environmental conditions experienced in The Great Plains. Traditionally these plots are subjectively screened visually for agronomic and performance traits that dictate the advancement into the breeding program. Visual scoring is subjective and introduces inter-rater bias and not the most robust method to produce quality data. Recent advancements in high-throughput phenotyping (H-TP) has allowed the use of imagery analysis to replace traditional scoring methods with more robust and repeatable methods of accurately quantifying agronomic traits like FI.

Objectively, H-TP paired with deep learning models like convolutional neural networks (CNN) can accelerate the rating of FI severity on research plots and identify cold-tolerant varieties that would be ideal for advancement into commercialization. Developing a deep learning method to evaluate FI in winter wheat is novel approach and the datasets are rare to collect due to the randomness of when freeze events occur. There is much interest in this work from the UNL agricultural engineering department as well as the UNL small-grains breeding program to better understand FI and develop advanced models for improving breeding methodologies.

Approaches and Data set

The FI dataset consists of 4,000 individual plot images ranging from healthy to severely injured winter wheat plots with annotated severity ratings (1-9), where 1 is healthy and 9 is severely injured.

The objectives of this study are the following:

1. Train a number of differing deep learning CNN models to predict FI.
2. Compare models and evaluate which generalizes the best for FI.
3. Suggest future work for further developing this research.

1.4 Project Idea 3: Clinical Diagnosis Based on the NIH Chest X-ray Dataset

Chest X-rays are a cost-effective and frequently medical imaging examination. But, it is hard to establish a clinical diagnosis using chest X-rays compared to other expensive medical imaging techniques such as CT imaging.

The goal of this project is to achieve clinically relevant computer-aided detection and diagnosis using chest X-rays with convolutional neural networks.

The dataset includes about 112,000 X-ray images with disease labels from about 30 thousands patients. The labels were created using text extraction from radiological reports [27]. Hence, the labels are not 100 percent accurate and are suitable for weakly-supervised learning.

The X-ray images can belong to one or more of the following 15 classes: Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural_thickening, Cardiomegaly, Nodule Mass, Hernia, and No findings.

This project is interesting because several intensive and emergency care units across the United States rely on X-ray imaging diagnosis to decide on the best treatment in a time-sensitive context.

Unfortunately, besides the low accuracy of the provided true labels, only about one thousand images come with predefined bounding boxes. The detection of regions of interest will be unguided for the most part, which can increase training time and impinge on the convergence.

1.5 Conclusions

The topics are presented in the order the group is interested in them. The natural language inference project seems to be the most interesting and has a complete dataset, but the group is unsure which model will be the best fit. Evaluating freeze injury in winter wheat has be partially done by one of the group members for their research. But the dataset is not a large as the group would like. The project may be expanded to additional crops to compensate. The third project is a more classical project in convolutional neural networks. There are multiple data sets available for this project and models that can be adapted for this project.

Chapter 2

Milestone 2: Project Selection

The group decided to work on a Natural Language Inference (NLI) project. The goals of NLI is to represent the meaning of a sentence as a vector. This will be useful for question answering, information retrieval and extraction, and text summarizing. The project will focus on single sentences but can eventually be scaled to paragraphs or entire documents. Our specific project will input a statement and a hypothesis and the output will be if they support each other, if they are neutral to each other, or if they contradict each other.

2.1 Introduction

Natural Language Inference (NLI) task is falls under the purview of Natural Language Processing (NLP). Recognizing the notions of entailment and contradiction are core to the process of natural language understanding. Thus, modelling inference can further enhance semantic representation of language.

In this task, we try to infer the relationship between a pair of sentences using a deep learning model. When such a task is presented to a person, the relationship between sentences is inferred from the meaning of words or semantics. In a neural network model, this semantic modelling is achieved using two mechanisms, attention and language representation. At a high level, this process entails two tasks to be performed by a deep learning model. First, every word in a sentence is represented numerically such that the numbers associated with a word represent it's similarity to other words in the sentence. Additionally, each sentence is parsed by the model to look for important words, or terms that it needs to pay 'attention' to. This step is done for both premise, and the hypothesis. Once such words are identified, the model measures the similarity between the key terms from the pair of sentences and predicts the relationship between them.

As stated before, this task can be broken down into two sub tasks. The first

one aims at capturing the meaning of words through language representation. This can be achieved by creating a vocabulary of all words present in the corpus or data set and representing them as vectors that capture the relationship of a particular word with respect to the rest of the words in the vocabulary. However, this is a naive approach that has been proven insufficient to capture the complexities of natural language due to its vast nature and inherent complexities. Thankfully, research in the area of language representation has led to the development of pre-trained models such as GLoVE [3], Word2Vec [20] etc that could be used directly as an embedding layer or could be fine tuned for the task at hand.

Furthermore, research has shown that using pre-trained models on a task similar to the one being undertaken help the model achieve better performance on downstream tasks. For this reason, using pre-trained models such as BERT [12], and ELMo [21] that have been trained for question-answering or auto-summarization tasks are a good candidate to use for the inference task. Moreover, LSTM and transformer based architectures are better suited to handle natural language processing tasks due to their ability to retain context while working with sequential data. Thus models that are LSTM based (ELMo) or transformer based (BERT) have attention mechanism included making them suitable candidates for the inference task.

While creating an accurate semantic representation of natural language is a complicated task in itself, progress in this area started with the introduction on generic pre-trained word representations. Research in this area continues as pre-trained models for specific tasks have been introduced to facilitate building better models and reducing the time required to train such models through transfer learning. At present, there does not exist a pre-trained model that is trained on an inference task. One of the impacts of this project is availability of a pre-trained model that is trained specifically for inference task. Furthermore, such a model could be extended to provide explanations for inference or fine tuned for domain specific inference.

2.2 Problem Specification

This project aims at building a fast, efficient and accurate model that learns language representation to apply it on an inference task. The motivation behind taking up this challenge is to

1. Build a model that effectively learns the representation and not just relies on artifacts present in the data.

2. Create a model that could be fine-tuned for inference on a domain specific task such as scientific data or clinical data.

The dataset consist of a pair of sentences called premise and hypothesis. The task is to predict whether the hypothesis entails or contradicts the premise. Three types of inference labels are possible, 'entailment', 'contradiction', 'neutral'.

The project will be developed in two phases. The data set that we would be using for training, validating, and testing during the first phase is the Stanford Natural Language Inference dataset [11]. Based on the leaderboard published at the Stanford NLP website [7], the top test accuracy achieved by a deep learning model on this dataset is 92.1. We aim to build a model that attains a test accuracy within the top 20 range.
Phase two consist of testing the quality of language representation of the model. The proposed plan is to fine tune the model on the Heuristic Analysis for NLI Systems (HANS) dataset [19], which is a controlled evaluation set containing examples that do not contain any artifacts or heuristics that could be used by a model for inference label prediction. Performance of our model on a challenging dataset like HANS will help us evaluate the learned representations of our model better.

Our experiments would be performed on crane and will use libraries such as Numpy and Pandas for data processing. Seaborne and Matplotlib for data visualization, Transformers and Tensorflow Hub for pre-trained models, and Keras and Tensorflow for building model layers.

2.3 Proposed Method 1: Long Short Term Memory

The Long Short Term Memory (LSTM) technique helps alleviate the vanishing gradient problem for longer phrases is text based recurrent neural networks. The structure of a LSTM cell is shown in 2.1 [14].The inputs are the current data input x(t), the previous state long term memory c(t-1), and the previous state short term memory h(t-1). The goal of the LSTM cell is to have a sense of long term memory of the data throughout the RNN and to have a short term memory of more recent data.

For the forget signal f(t) the previous short term data, h(t-1) and the current input data x(t) are input into a logistic function whose output is then multiplied by the previous state long term memory c(t-1) at the forget gate. This "forgets" some of the previous long term data. The next gate, g(t) is the main gate that takes h(t-1) and x(t) through a tanh function to have the new data that is added to the previous long term memory. Before g(t) is added to the long term memory it is multiplied by i(t), which is similar in activation to f(t), which decideds how much to g(t) is added to the long term memory. That long term memory is passed to the next LSTM cell. This long term memory is sent through another tanh function then multiplied by o(t), which serves the same function as i(t) in that it decides how much information is passed to the short term memory h(t). The output of the cell y(t) is equal to the short term memory output h(t).

The group plans on using a model including LSTM's among other techniques to achieve high validation accuracy on different NLI corpus's. The main data set is the Stanford natural language inference corpus, with the multi-genre natural language inference corpus and the SciTail entailment dataset also being

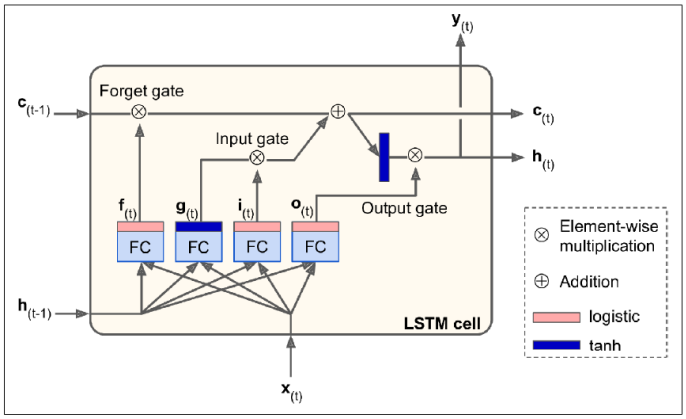

Figure 2.1: LSTM cell block diagram.

applicable.

The performance of our model can be compared with a scoreboard of notable NLI models found at [5]. Also, notable models can be used for comparison on inspiration [26], [24].

2.4 Proposed Method 2: Reusing Embeddings Pretrained Through Self-Supervision

This approach will use pretrained embeddings from a Biredirectional Encoder Representation from Transformers (BERT) Tensorflow module hosted at the TensorFlow Hub project.

Each word or sub-word appearing in a text has to be treated as an individual text token for a model to understand an unstructured text. Thus, each token's representation can be pretrained on a much larger corpus, such as word2vec, GloVe, or other subword embedding models.

Upon pretraining, words or tokens can have a vector representation, allowing a model to learn and recognize particular patterns [29].

However, the vector representation remains the same no matter what the context is. For instance, the vector representation of "state" is the same in both "the state of Nebraska" and "it is in a good state."

Recent start-of-the-art (SOTA) models attempt to provide a context-based vector representation of tokens or words in a sentence. BERT [12], based on the Transformer Encoder, is amongst such SOTA architectures. Figure 2.2 depicts how BERT fits into our second approach.

In the next two weeks, we will explore a range of BERT modules from the Tensorflow Hub project to find which can be adequately fine-tuned for the SNLI task.

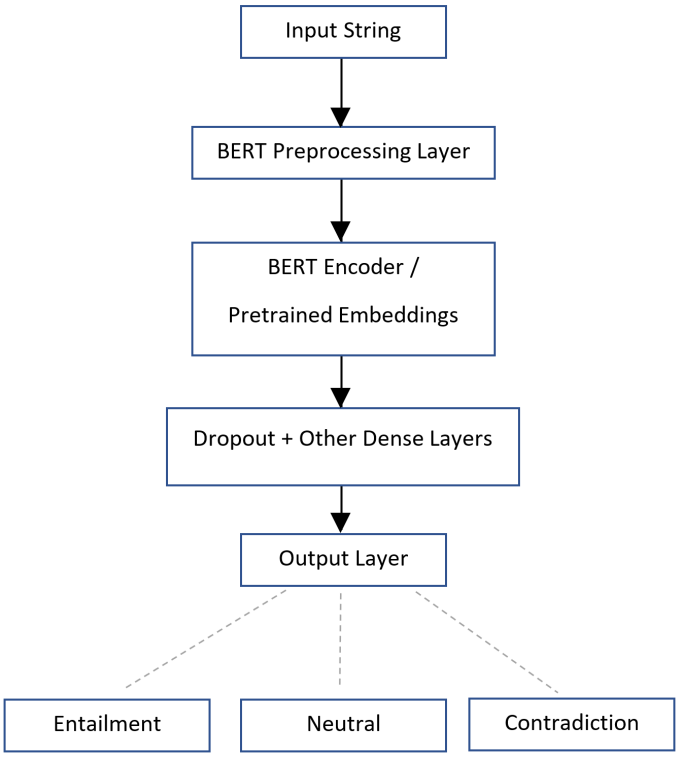

Figure 2.2: High-level representation of an architecture reusing embeddings pretrained on large corpora through self-supervision with BERT or BERT-like models. Adapted from [2, 8].

Some of the BERT models we will explore are described by [1, 8] and include: the BERT-Base modules, which BERT authors initially released; the Small BERTs, which might be suitable to assess a trade-off between speed, size, and quality of predictions; ALBERT or A Lite BERT, which allows parameter sharing across layers thus reducing the model size; BERT Experts, a series of eight BERT modules providing specific embeddings based on the task; etc.

By the third milestone, we shall report on the BERT module that provided the task's best performance.

In the next steps, we will try out different hyperparameter values through cross-validation. We shall also tweak the architecture in a bid to attain a performance that may rank among the top based on the benchmark at the SNLI web page.

2.5 Conclusions

The goal of using natural language inference is to represent sentences as vectors. This project will compare those vector together and decide if they support each other, or neutral to each other, or are contradictory. With enough progress similar models can represent paragraphs or entire documents that can be used for question answering and summarizing. For model architecture, LSTMs, GRUs, and other RNN techniques will be explored since they have been used to different degrees of success in literature for models on the scoreboards [[**?**]]. Similar models can also be leveraged using BERT, which will hopefully yield high accuracy with lower training times.

Chapter 3

Milestone 3: Progress Report 1

3.1 Introduction

The group is working on providing a fast and accurate model to predict semantic relationships between a pair of statements, or commonly known as natural language inference (NLI). The goal is to infer the relationship between two sentences with a deep learning model. The model predicts whether the two statements, the premise and the hypothesis have an entailment, contradictory, or neutral relationship, whereby the hypothesis truthfulness is dependent on the premise. The output is zero for entailment, one for neutral, and two for contradiction. The dataset used for training, validation and testing is the Stanford natural language inference dataset (SNLI) [10] and it was retrieved from the Tensorflow Datasets collection.

The group is using transfer learning from bidirectional encoder representations from transformers, or BERT, for NLI [13]. A BERT model is imported from tensorflow hub, and with additional preprocessing layers and output layers the BERT model is trained on the SNLI dataset. The BERT model is very large, with the best MultiGenre NLI model having close to 110 million parameters and requires a significant amount of training time. Our goal is to use smaller versions of BERT with additional input and output layers to achieve a high accuracy with the SNLI data set, and perhaps other data sets, while maintaining a reasonable training time.

For this milestone, the group used three different models of BERTs, a small, medium, and large. As expected training time increased with model complexity but non linearly. Also, test accuracy increased with model complexity but again did not scale linearly. Results are discussed further in 3.3.

Table 3.1: Other Network Parameters for Search Space

Models	L	H	A	Trainable Parameters	Non-trainable Parameters
Tiny BERT	2	128	2	99587	4385921
AlBERT	24	1024	16	262659	17683968
MNLI-BERT	12	786	12	427267	109482241

3.2 Experimental Setup

The SNLI data set is obtained from [10]. It includes 500,000 training examples, 10,000 validation, and 10,000 test. The preprocessing layer is taken from [13]. The preprocessing tokenizes the input sentences, or splits it into individual words and word parts. The tokenizer uses 'wordpiece tokenization'. This is based on a sub-word segmentation algorithm popular in NLP tasks. This algorithm retains frequently used sub-words while splitting the infrequently occurring words. For example, the word 'annoyingly' is split into tokens 'annoy', 'inly' since there are far many words that end with 'inly' than the original word. This kind of tokenization helps in limiting the size of the corpus vocabulary which in turn makes the tokenization process efficient and economical. The inputs to the BERT model are the input words, the segment or phrase embeddings, and the position embedding of where the word is in the dataset [16]. The maximum sentence length used is 128, with an embedding dimension of 768.

The BERT models are loaded from Tensorflow hub. Three different BERTs are tested according to their size of parameters: small, medium and large. The small model, or Tiny BERT, has two layers, with a hidden size of 128, and 2 self-attention head [25]. It is a trimmed down version of the original BERT model. The medium model is an AlBERT model [18]. AlBERT is similar to BERT, but it can be trained faster due to the vocabulary being split into two smaller matrices and layers sharing parameters. It has 24 layers, a hidden size of 1024, and 16 attention heads. The large model is the original BERT that was fine tuned on the MNLI dataset for 12 epochs [13]. It has 12 layers, a hidden size of 786, and 12 attention heads. A summery of the models is shown in Table 3.1.

The output of the BERT is passed to a bidirectiononal LSTM layer with 64 units. The LSTM layer is then used as an input to a max pool and average pool layer that are concatenated. The outputs of this layer are then dropped at a rate of 30% as they are passed to the output layer. The output layer is 3 dense neurons that have softmax activation. The three neurons are for each of the possible outputs.

A batch size of 32 is used to help limit training time. Adam optimization with an initial learning rate of 1e-5 is used. For each model two epochs are trained without training the imported BERT model, then two epochs are trained with training the BERT model, or fine tuning. The first two epochs allow the non-BERT layers to be trained, while the last two epochs fit the entire model to the problem space. The accuracy and the training time for each model is measured.

Table 3.2: Other Network Parameters for Search Space

Models	Total Parameters	Training Time	Test Accuracy
Tiny BERT	4,485,508	1.9 hr	77%
AlBERT	17,946,627	2.2 hr	85%
MNLI-BERT	109,482,241	2.8 hr	90%

3.3 Experimental Results

As expected, tiny-BERT performed the fastest and the least accurate while the MLNI-BERT model took the longest to compute and had the most accurate results. The results are summarized in Table 3.2.

3.4 Discussion

The test accuracy vs. model parameters is shown in Fig 3.1 and the computation time vs. model parameters is shown in Fig 3.2. The MNLI-BERT increased the accuracy of the Tiny-BERT model by 13% but took 47.4% longer to compute. The AlBERT model increased accuracy by 7% and only increased computation time by 15.8%. This leads one to believe that the methodology behind the AlBERT model decreases training time while maintaining high accuracy. The additional training on the MNLI-BERT model on a similar dataset likely improved the accuracy.

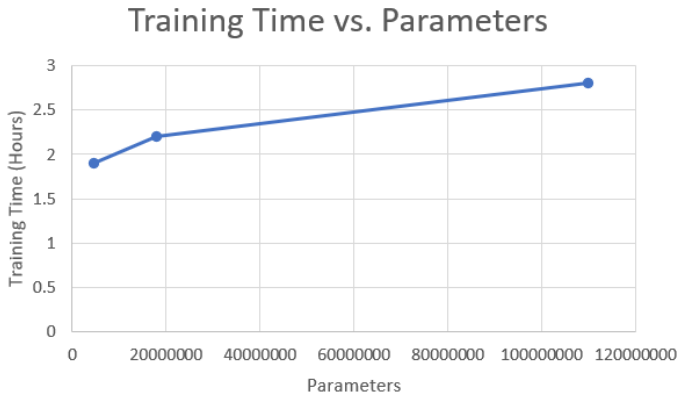

Figure 3.1: Training time of the models for 4 epochs vs. number of parameters

3.5 Conclusion

The group has used transfer learning from BERT models on Tensorflow datasets for natural language inference on the SNLI dataset. The group is trying to find

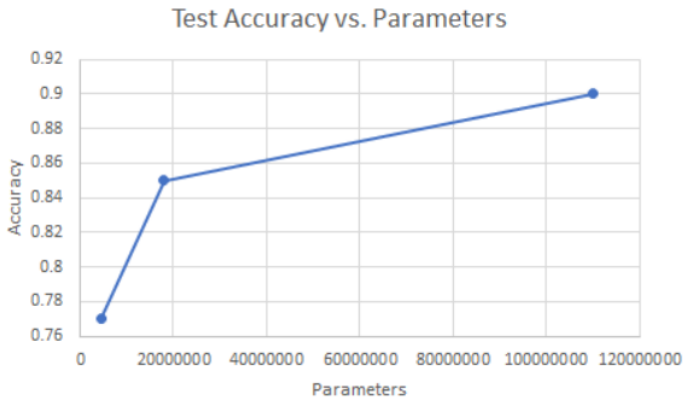

Figure 3.2: Accuracy of the models vs. number of parameters

the right combination of BERT model and additional layers to produce the most accurate network while achieving fast computation times.

AlBERT produced results that were better than expected. The group will begin looking at other improvements to the BERT model: RoBERTa, ERNIE, and DistilBERT. The group is also planning on doing a grid search of output layers to find the best in terms of accuracy and training times. Other output layers like convolutional layers and GRUs will be explored. The group will also consider training the model on a portion of the training data, instead of all 500,000 samples.

When importing models, there appears to be compatibility issues between Tensorflow, numpy, and other modules. The group is working on overcoming some of these issues. The projects usually work in Google Colab, but the group has problems when trying to run the code on HCC.

Overall, the group plans to provide a somewhat comprehensive list of available BERT models and implementations measured by accuracy and training time for the SNLI dataset.

Bibliography

[1] URL: https://tfhub.dev/.

[2] Classify text with bert nbsp;: nbsp; tensorflow core. URL: https://www.tensorflow.org/tutorials/text/classify_text_with_bert.

[3] GloVe: Global Vectors for Word Representation. URL: https://nlp.stanford.edu/projects/glove/.

[4] Natural language inference on snli. https://paperswithcode.com/sota/natural-language-inference-on-snli. Accessed: 2021-02-10.

[5] Nlp-progress. http://nlpprogress.com/english/natural_language_inference.html. Accessed: 2021-03-4.

[6] Snli. https://nlp.stanford.edu/projects/snli/. Accessed: 2021-02-10.

[7] The Stanford Natural Language Processing Group. URL: https://nlp.stanford.edu/projects/snli/.

[8] Martín Abadi and et al. Ashish Agarwal. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

[9] Douglas S. Alt. Managing risks of soft red winter wheat production: Evaluation of spring freeze damage and harvest date to improve grain quality. Accessed: 2021-02-11.

[10] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 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, 2015.

[11] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. `arXiv:1810.04805`.

[14] Aurelien Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, 2017.

[15] Abdul Khaliq Umair Ashraf Shakeel A. Anjum Shengnan men Longchang Wang Hafiz A. Hussain, Saddam Hussain. Chilling and drought stresses in crop plants. *Front. Plant Sci. 9:393. doi: 10.3389/fpls.2018.00393*, 2018.

[16] Rani Horev. Bert explained: State of the art language model for nlp, Nov 2018. URL: `https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270`.

[17] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[18] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020. `arXiv:1909.11942`.

[19] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[21] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[22] Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*, 2018.

[23] Shane Storks, Qiaozi Gao, and Joyce Y Chai. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*, 2019.

[24] Aarne Talman, Anssi Yli-Jyrä, and Jörg Tiedemann. Sentence embeddings in nli with iterative refinement encoders. *Natural Language Engineering*, 25(4):467–482, Jul 2019. URL: http://dx.doi.org/10.1017/S1351324919000202, doi:10.1017/s1351324919000202.

[25] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models, 2019. arXiv:1908.08962.

[26] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. arXiv:1804.07461.

[27] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[28] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[29] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2020. https://d2l.ai.