Investigating the Limits of MAML Meta-Learning for Few-Shot Learning in a Bi-LSTM Model

Tommaso Rivetti
Cognitive Science,
McGill University

Mika Desblancs
Computer Science and Philosophy,
McGill University

Eliott Regnier
Computer Science,
McGill University

Abstract

Deep neural networks are known for their excellent performance on classification tasks in NLP. However, a large amount of training data is required in order to generalize to generalize to unseen data and achieve competitive performance. On the other hand, simpler ML models such as Logistic Regressions or Naive Bayes models start giving accurate results on classification tasks with much less input data in training. This paper seeks to experiment with few shot learning through the use of a Model Agnostic Meta Learning model (MAML) to bridge the gap between low training data size and performance. First, we implemented the MAML learning algorithm, built a Bi-LSTM model and train the model first normally, then using the MAML algorithm. Then, we built a Naive Bayes- and Logistic Regression- classifier. We then compared how many training points the MAML-augmented Bi-LSTM model needed to see before it got results similar to the simpler models on a binary classification task. The goal is to determine how few datapoints are required for few shot learning augmented deep neural model to surpass simpler models. We find that a Logistic Regression model with GloVe word embeddings still performs better than our MAML augmented Bi-LSTM on low amounts of data. However, the latter does in fact yield high results (90% and up on F1) after seeing a smaller number of points than a vanilla Bi-LSTM model. Although it may not be quite competitive enough with the Logistic Regression model on few samples, our model can still be optimized further.

1 Introduction

1.1 Motivation

This project aims to find out whether a neural network (NN) model can be used for a range of classification tasks given that it has seen few data points. The motivation behind this project stems from the idea of attributing authorship to written texts that may be short in length and not provide much information for identification. For example, anonymous harmful comments or social media platforms can be posted, and we might want to attribute a person or username to the comments, but only have access to very limited “works” of theirs. While we did not work on author attribution, we hope the work provided here could lay the groundwork for future work in this area. In this experiment we investigated few shot learning: “making predictions based on a limited number of samples [...]” where the goal is not to let the model recognize the labels in the training set and then generalize to the test set, but instead learning to learn” [1]. This is also known as meta-learning, where the goal is to create models capable of “rapid adaptation” to new tasks. Our project aims to investigate the threshold at which a NN model is able to outperform simpler models, and we test this through the use of MAML-augmented Bi-LSTM and increasing the number of points it sees by batches of **16** points.

1.2 Introduction to MAML

The authors of *“Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”* have created the Model-Agnostic Meta-Learning meta-learning algorithm. Its purpose is to find a model’s optimal parameters so it can learn new tasks quickly. The algorithm is model-agnostic in that any model trained with gradient descent is MAML compatible. In the algorithm, a model is “trained during a meta-learning phase on a set of tasks, such that the trained model can quickly adapt to new ones”[2]. Formally, each task consists of a loss function, a distribution over initial observations, a transition distribution, and an episode length (the model generates samples of length = episode length). The

[1] Logeswaran, Lajanugen, et al. Few-Shot Sequence Learning with Transformers. 2020.

[2] Finn, Chelsea, et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 2017.

meta-learning allows the model to “learn a distribution over the tasks”. The intuition behind the approach is that in the meta-training phase the model learns internal representations which can be transferred for fast adaptation. The model’s parameters are thus set such that they are sensitive to training on a new task so it can be easily generalized and adapt quickly.

1.3 Previous Works and Hypothesis

The MAML paper we discussed provided the algorithm we built this experiment upon. However, they use convolutional neural networks and focus on image classification tasks. In our experiment, we decided to work with text classification tasks and investigate how quickly the model could classify text. The dataset we wanted our model to learn quickly was inspired by this paper *“Automatic Detection of Sexist Statements Commonly Used at the Workplace”* [3] which explores different models to classify sexist comments. They build various Bi-LSTM models along with Logistic Regression and Naive Bayes models. Achieving F1 scores between 83 and 88 across their different models, their results were useful guidelines for the scores we wanted to see in our models. Furthermore, the authors state that since they only had $1,000$ data points, they believe the model could perform better if it had more data. We decided to use the MAML implementation on the Bi-LSTM model the paper described, to test its compatibility to the NN, and to discover how fast the model could learn to classify sexist comments in the workplace. The ”rule of thumb is that we need 1000 samples per class” [4] for a ML learner, let alone a NN.

Our hypothesis is that with a MAML-augmented Bi-LSTM model can learn a lot faster than a vanilla Bi-LSTM model and can achieve a competitive f1-score faster than non-neural network based models, trained on the number of points the MAML-augmented Bi-LSTM sees. (When we reference ’competitive’ we mean the model quickly achieves its maximum F1 score).

2 Method

2.1 Packages

Our experiments rely heavily on numpy, sklearn and nltk packages for preprocessing data and the “simple” model implementations. We used pandas, keras and pytorch for implementations of MAML and our Bi-LSTM. We also used gensim for global vectorization, and matplotlib for plotting information.

2.2 Datasets

The datasets used were the Sexist Workplace Statement, Amazon Reviews for Sentiment Analysis and IMDB Dataset of 50K Movie Reviews, all from Kaggle. The Sexist Workplace Statement dataset contains 1100 comments, where 55% are classified as sexist and 45% are classified as not sexist.The IMDB movie review dataset has $50,000$ reviews where 50% are positive and 50% are negative. Originally, the Amazon dataset contained more than 2 classes. Since MAML assumes the same number of classes for the training and target datasets, we picked only two classes from the Amazon data. Furthermore, to prevent overloading RAM we shortened the dataset. The final dataset, Mini Amazon Reviews has $62,500$ reviews with 50% positive and 50% negative reviews.

2.3 Preprocessing and Feature extraction

For the non-neural network models we experimented with various pre-processing steps in order to keep only those which gave the best results with the Naive Bayes and Logistic Regression classifiers. The candidate pre-processing steps were special characters, setting all words to lowercase, de-punctuation, lemmatization and clearing of stop words. All steps were performed using nltk and the wordnet package. For both the Logistic Regression and Naive Bayes models we use as input on n-grams of raw term frequencies, term frequency–inverse document frequency (TF-IDF) vectorization or the averages of the GloVe word embeddings for each word in the sentence. N-grams range for N ranging from 1 to 5.

For the Bi-LSTM model we simply lower-cased, de-punctuated and removed special characters for the sentences in the dataset. Furthermore, we used the GloVe word embedding word representations. Finally, in order to deal with sentences of varying length we simply pad the input sentences so they

[3] Dylan, and Patricia Conde-Cespedes. Automatic Detection of Sexist Statements Commonly Used at the Workplace.

[4] Melvin, Author Ryan L. “Sample Size in Machine Learning and Artificial Intelligence.” Perioperative Data Science, 8 July 2021.

all have the same size when fed into the Bi-LSTM model.

2.4 Models

We created a Logistic Regression (LR) and a Naive Bayes (NB) model to use as controls for our experiment, and a Bi-LSTM RNN to be used with MAML. We took inspiration from the paper *"Automatic Detection of Sexist Statements Commonly Used at the Workplace"* [5] for our implementations of the LR and Bi-LSTM architecture. Their paper compares an LR and a Bi-LSTM model on the same dataset, both enhanced with GloVe word embeddings. We kept similar model architectures to theirs so as to know what to expect in terms of performance on classification tasks.

2.5 Logistic Regression

Our logistic regression model uses the sk-learn logistic regression implementation, taking either a vectorized N-gram (range 1-5), TFIDF, or averages of the GloVe word embedding for each work in the sentence as input after passing our training data through the pre-processing steps we mentioned previously. We tested on all datasets but kept results for the Sexist Comments one to compare with the MAML Bi-LSTM.

2.6 Naive Bayes

Similarly, our Naive Bayes model uses the Gaussian Naive Bayes implementation from sk-learn, and takes either the N-gram, TFIDF or the average of the GloVe word embeddings for each word in the dataset. Again, we tested on all datasets but kept results for the Sexist Comments one to compare with the MAML Bi-LSTM.

2.7 Bi-LSTM

The architecture consists of a 50 dimensional embedding layer created using GloVe, then 2 bidirectional LSTM layers. We then concatenate the hidden cell state of the forward LSTM at time-step t_n and the hidden cell state at time step t_0 of the backward LSTM in the last bidirectional LSTM layer before feeding them into a fully connected layer with an input size 64 and output of size 1. This is then fed into a soft-max activation function. We use the Binary Cross Entropy loss function and the Adam optimization algorithm [6]. A vanilla version of the model was run to compare with a MAML augmented Bi-LSTM.

2.8 MAML

We used the mini Amazon Review dataset and IMDB movie reviews for the different binary classification tasks that are used for training and from which we sample from in the inner-loop. However, there is a different embedding matrix for each task because they have their own specific vocabulary. Therefore, every time we sample from the tasks, we load parameters from our learner model, save them, and then create a new learner with those same parameters but with the word embedding layer adapted to the task we are sampling from. In terms of hyper-parameters, we used: the number of times we sample for each dataset was 2 (inner update), the loss function we used to evaluate tasks was the Binary Cross Entropy loss function, the inner step size was 0.01 (task specific gradient update learning rate), the meta-gradient step size is 0.01, and the number of meta-updates was 30. We used a training batch size of 8 and validation size of 128 for both the Amazon and IMDB. Validation batch sizes were of 128 for all datasets. To determine which MAML hyper-parameters were best we ran the outer-loop multiple times for each combination of hyper-parameters and kept track of the F1 scores, picking the hyper-parameters which seemed to produce the model which learns the quickest.

2.9 Experiment Design

Our experiment was divided in two parts. First we built various non-neural models with Logistic and Naive Bayes classifiers which differ in the pre-processing steps. Then, we trained the non-neural models on dataset sizes ranging from 16 to 880. At every training set size we measure the model's F1 score on the test set. Then, we train a vanilla Bi-LSTM model by letting it see 16 data points at a time for 5 epochs and track the model's score on the training set every time we let it see 16 datapoints. Note that we aren't training the neural network model on training sets sizes ranging from 16 to 880. Rather, we calculate the model's F1 score every time it sees another 16 datapoints. Finally, we repeat the procedure for a MAML-augmented Bi-LSTM model. We compare the F1 scores at every 16 points and draw conclusions.

[5] Dylan, and Patricia Conde-Cespedes. Automatic Detection of Sexist Statements Commonly Used at the Workplace.

[6] Adam: A Method for Stochastic Optimization

3 Results

We ran iterations of increasing sample sizes for the training set of the non-neural network models, and plotted them against F1 for each model. For our non-neural network models, we experimented with the previously described pre-processing and feature extraction steps but kept only the best performing models for comparing to our Bi-LSTMs. The training size range was 16 to 880 with a step size of 16 for LR. For NB, the range was 16 to 880, with a step size of 16. The GloVe models performed best on low samples for both LR and NB as can be seen in Figures 1 and 2. We therefore kept those and plotted them against the deep NN.

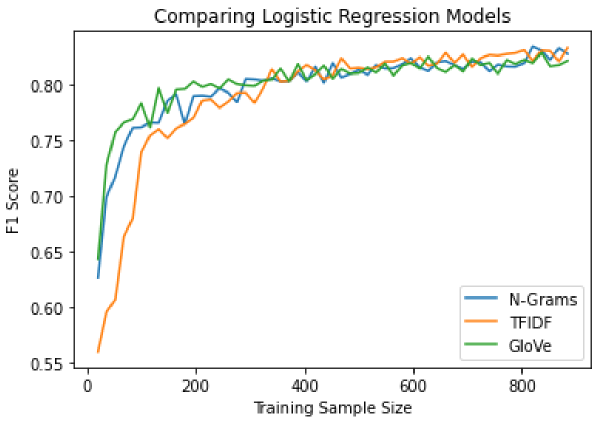

Figure 1: Comparing the F1 Scores of different Logistic Regression models (GloVe, TFIDF, N-Grams) with respect to the number of data samples used for training

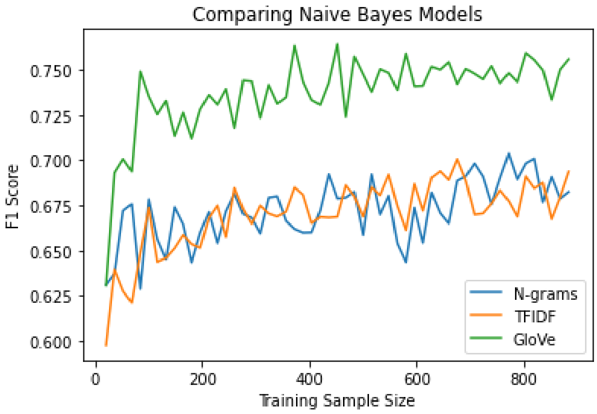

Figure 2: Comparing the F1 Scores of different Naive Bayes models (GloVe, TFIDF, N-Grams) with respect to the number of data samples used for training

Then we trained the Bi-LSTM and MAML-Augmented Bi-LSTM model by letting it see 16 points at a time for 5 epochs. Saving the F1 score on the validation dataset at every 16 points and averaging the F1 scores on 10 different runs we get Figure 3. From Figure 3 we see that while both models reach an F1 score of 80 almost at the same time, the MAML-augmented model goes above 80 before 1,000 data points are shown and continues to climb in to achieve 87% on average after 5 epochs. On the other hand, the vanilla Bi-LSTM never surpasses the MAML-augmented Bi-LSTM after the first 1,000 data points are seen and it's average score on the validation dataset is wildly inconsistent compared to the continuous increase of the MAML-augmented Bi-LSTM.

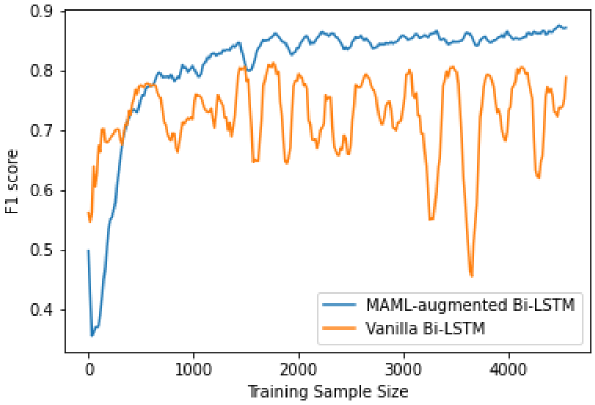

Figure 3: Comparing the F1 score on the validation dataset of the MAML-Augmented and vanilla Bi-LSTM

Comparing the Bi-LSTM models to the non-neural methods on the first 880 data points however, we see that the simpler models clearly outperform both the vanilla and MAML-augmented Bi-LSTM models, on the first 200 data points. With the GloVe Naive Bayes model reaching an F1-score of 70% and GloVe reaching Logistic Regression getting an F1 score of 80%. The MAML-augmented Bi-LSTM surpasses both the GloVe NB and vanilla Bi-LSTM at the end of the 880 data points. From Figure 1 we see that the GloVe Regression model reaches its max F1 score of 83% after 600 points. In Figure 3, however, we see it takes the MAML-augmented Bi-LSTM about 1200 points to reach an F1 score of 83%.

However, from the results shown above, the simpler models always performed better on smaller training sets and the neural models weren't able to out perform the simpler models after being shown a small number of data points once.

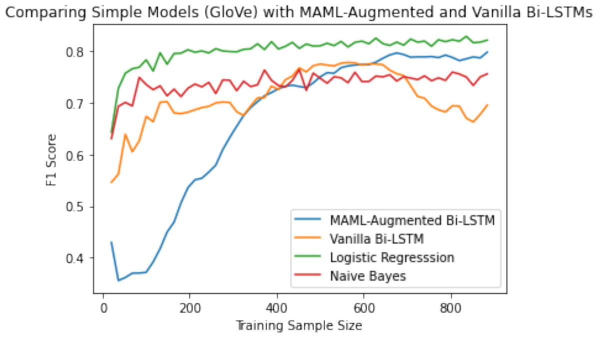

Figure 4: Comparing The F1 scores of the top performing Logistic Regression, and Naive Bayes models (GloVe) against the Vanilla and MAML-Augmented Bi-LSTM Neural Networks with respect to the number of data samples used for training

4 Discussion

Given our results, we can tentatively validate our hypotheses that the MAML-augmented Bi-LSTM can learn faster than vanilla Bi-LSTM. However, it does not outperform simpler methods after being shown small amounts of data points while the simpler models were trained on these small numbers of points. Training our Bi-LSTM with MAML does seem, as well, to have a stabilization effect regarding a model's score on the validation dataset. While the vanilla model's score on the validation data fluctuates a lot, the MAML-augmented Bi-LSTM's scores continuously rise and stabilize around its optimal score parameters. Thus, MAML might have a stabilizing effect on the model it trains.

4.1 Issues

We ran into several issues concerning preprocessing and word embeddings using GloVe. RAM usage was too high and led to several crashes, but this was addressed by reducing the dataset sizes. Very large datasets were beyond the scope of our experiment. Finally, we ran into many issues concerning the optimal MAML hyper-parameters. The algorithm requires computationally expensive hyperparameter searches in order to find the optimal version for a given set of tasks. The hyper-parameters we tune are the inner loop learning rate, outer loop learning rate, number of updates in inner loop, and number of times you run the outer loop. Our program crashed numerous times when testing for different hyper-parameters because of RAM overload and session timeouts. Furthermore, we had planned on running the experiments on more datasets than just the sexist comments one. However, due to time constraints and limited computational power we were unable to do this.

4.2 Project Extension

In terms of the NN we used, we would have liked to use a Transformer model because they are very successful at modeling discrete sequences and have received a lot of attention in recent academic papers. They have been shown to use context tokens appended to an input to adapt their generations or to switch between different tasks. This can be effectively harnessed in the meta-learning settings for few-shot learning. ("Few-Shot Sequence Learning with Transformers" by L Logeswaran). The problem with transformer models is they require very high computational power in order to train or update them in a meaningful way because of how many parameters they have. This could not be achieved within the scope of this paper. However, computational power permitting, we could extend our experiment by using the Leopard few shot learning algorithm with a pre-trained Bert model as the encoder to test a model with even greater performance with few samples. The paper mentions it showed impressive results on 17 different classification tasks and used 4 GPUs [7] in their updates of BERT parameters. We had to limit ourselves to training on a much smaller number of classification tasks. Furthermore, given that the biggest challenge we faced was the MAML hyper-parameter search, we would like to develop heuristics and methods to find MAML hyper-parmeters. Finally, we believe it would be interesting to investigate the stabilizing effect of MAML we picked up in our experiment.

5 Statement of Contributions

Mika: Implementation of MAML and the Bi-LSTM, including its training, testing and hyper-parameter search, writing of report.
Eliott: Implementation of Naive Bayes and Logistic Regression models, including training and testing. Gathering and Plotting Results
Tommaso: Preprocessing, contribution to Logistic Regression training, writing of report

[7] Bansal, Trapit, et al. Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks.

References

[1] Bansal, Trapit, et al. Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks.

[2] Dylan, and Patricia Conde-Cespedes. Automatic Detection of Sexist Statements Commonly Used at the Workplace.

[3] Finn, Chelsea, et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 2017.

[4] Melvin, Author Ryan L. “Sample Size in Machine Learning and Artificial Intelligence.” Perioperative Data Science, 8 July 2021.

[5] Logeswaran, Lajanugen, et al. Few-Shot Sequence Learning with Transformers. 2020.