CS60075: Natural Language Processing Project - DSML Fanatics

Anonymous ACL submission

Abstract

Sign language is the only tool of communication for the person who is not able to speak and hear anything. Sign language is a boon for the physically challenged people to express their thoughts and emotion. In this work, a novel scheme of sign language recognition has been proposed for identifying the alphabets and gestures in sign language. With the help of computer vision and neural networks we can detect the signs and give the respective text output. We can predict the intent of produced text using novel Natural Language Processing Algorithms.

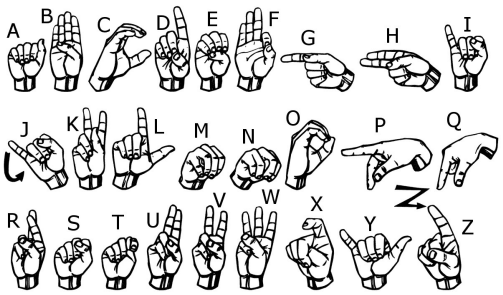
1 Introduction

In this work, we developed gesture recognition model which is trained on MNIST data set for identifying the alphabets and gestures in sign language. With the help of computer vision and neural networks we can detect the signs and give the respective text output. Then we predict the intent of sentence generated by gesture recognition model using intent classification model based on BERT which is trained on ATIS data set.

2 Sign Language and Hand Gesture Recognition

2.1 ASL Dataset

The data set is a collection of images of alphabets from the American Sign Language, separated in 29 folders which represent the various classes.The training data set contains 87,000 images which are 200x200 pixels. There are 29 classes, of which 26 are for the letters A-Z and 3 classes for SPACE, DELETE and NOTHING. These 3 classes are very helpful in real-time applications, and classification. The test data set contains a mere 29 images, to encourage the use of real-world test images.



signs

Figure 1: ASL Signs and their symbols

2.2 CNN - ResNet Model Architecture

The issue of disappearing gradients, which can happen when training deep neural networks, was the inspiration for us to use ResNet. By utilising skip connections, which let the network avoid one or more levels during training, it was able to achieve this.

We used ResNet to classify data and extract characteristics from the input in the context of ASL gesture recognition. It typically has an architecture made up of several residual blocks, each of which has several convolutional layers and shortcut connections. The network can learn residual mappings—the difference between a block's input and output—by using these short-cut connections.

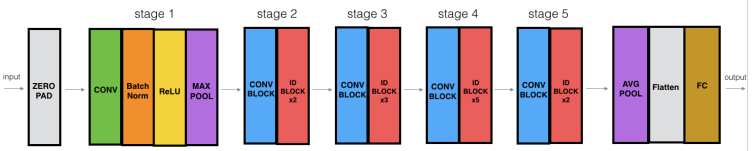


Figure 2: Our ResNet Based CNN Architecture

We have achieved very high accuracy of 99 percentage for the ASL problem statement.

3 Intent Classification

3.1 ATIS Dataset

Airline Travel Information System (ATIS) dataset It has been proven that spontaneous speech is built

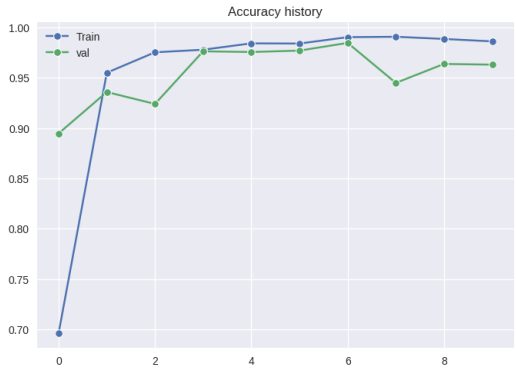


Figure 3: Accuracy Plot

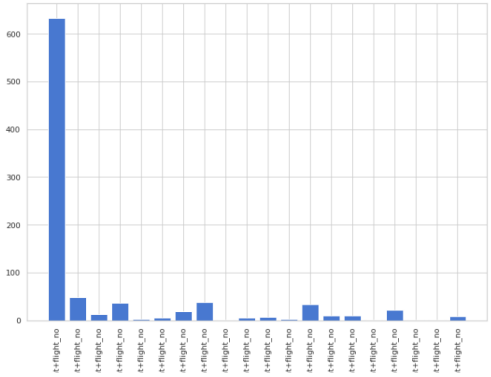


Figure 5: Intent Distribution over Testing set

on a corpus to be extremely valuable in understanding spoken language (SLU). In the training set of ATIS, there are 4978 samples With a vocabulary size of 943 as well as 129 different slots and 26 intentions. The test set consists of 893 samples, 129 slots and 26 intentions. Sentences have an objective distribution presented in Figure 4 and Figure 5.

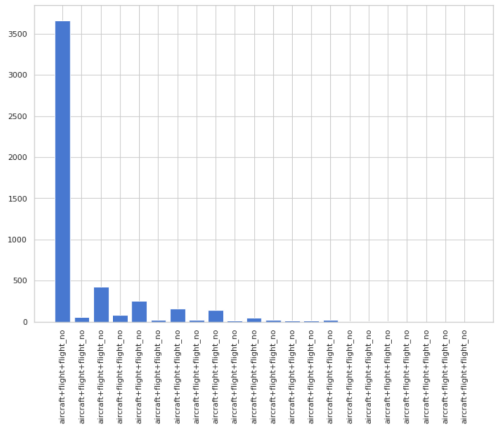


Figure 4: Intent Distribution over Training set

3.2 Preprocessing

The first step to train a model is to pre-process the data. This involves tokenizing all sentences and adding all sentences to match the data to the BERT input. Since BERT expects data to be tokenized with a start ([CLS]) and end ([SEP]) sentence, we use a tokenizer provided by BERT to convert the data into an acceptable format. Convert words to tokens as BERT vocabulary. The next step is to extract data based on BERT tokenizer vocabulary identifiers. All expressions are padded with the "[PAD]" token at index 0. Then split the data into training, validation and test sets with a ratio of 90:10.

3.3 BERT Model

BERT 4 is an open source pretrained transformer encoder stack model. It is the successor of OpenAI GT and reduces the limitations of sequencing models. BERT has served as the Swiss Army Knife of modern NLP. All previous models were only able to find contextual dependencies in one way. But BERT is completely bidirectional. It is constructively developed using the well-known clever tricks of language representation models. Available in different sizes, with different sized encoder layers and hidden dimensions. There are several versions, but the two standard versions are:

1. BERT-Large, Cased (Whole Word Masking): 24-layer, 1024-hidden, 16-heads, 340M parameters

2. BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters.

For this intent classification task, we used the second BERT-Base.BERT can be used for wide array of NLP tasks with one fine-tuned additional layer on top of actual model.

1. Sequence Classification Tasks

2. Sentence Pair Classification Tasks

3. Question-Answering Tasks

4. Single Sentence Tagging Tasks

For our project, we need to make minimal changes to one of our tasks called the Sequence Classification Task. We use attention masking where every word is a candidate for prediction, so we hide every cue. Add a classification layer using BertForSequenceClassification. Our model has a total of 13

layers: 12 layers for Attention Burt, Burt-Output, Burt-Average and 1 Classification Output Layer. Our loss optimizer is "Adam", developed by Adam Kingma et al. 15 , which combines features of the RMSProp and Adagrad algorithms to compute stochastic gradient descent (SGD) on first-order gradients. We use the first two moments to optimize different parameters.

3.4 Evaluation

This project was implemented in Jupyter Notebook through Google-Colab. ATIS data comes from Python pickle files. After the data was ready, We divided all the data into 32 categories and converted them into tensor iterators. Input the data into the model and train the model for 5 epochs. A graph of training drop for each period is shown in the following figure.

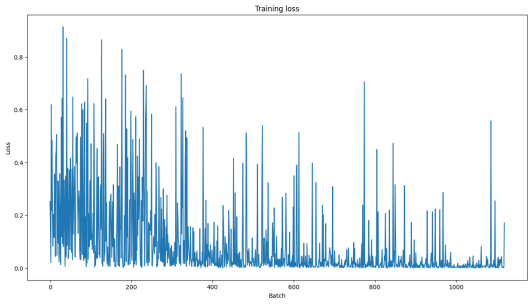


Figure 6: Training loss on all batches

After running the model on the test set, we used the correct and predicted labels to calculate the confusion matrix. I plotted this confusion matrix in a heatmap as shown in the figure below. We also created a classification table comparing metric scores in the figure below.

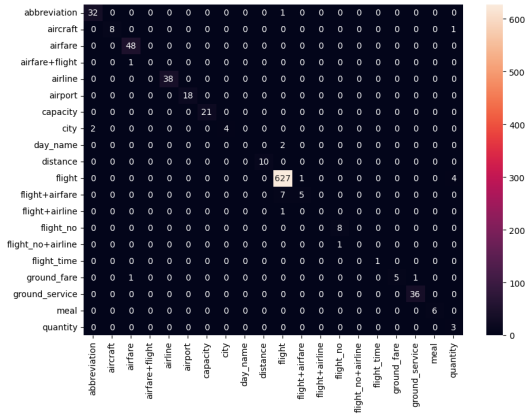


Figure 7: Confusion matrix

	precision	recall	f1-score	support
0	1.00	0.94	0.97	35
1	0.89	1.00	0.94	8
3	1.00	0.96	0.98	50
4	0.00	0.00	0.00	0
6	1.00	1.00	1.00	38
8	1.00	0.95	0.97	19
9	1.00	1.00	1.00	21
11	0.67	1.00	0.80	4
12	0.00	0.00	0.00	0
13	1.00	1.00	1.00	10
14	0.99	0.99	0.99	635
15	0.50	0.86	0.63	7
16	0.00	0.00	0.00	0
17	1.00	0.89	0.94	9
18	0.00	0.00	0.00	0
19	1.00	1.00	1.00	1
20	0.86	1.00	0.92	6
21	1.00	1.00	1.00	36
23	1.00	1.00	1.00	6
24	1.00	0.38	0.55	8
accuracy			0.98	893
macro avg	0.75	0.75	0.73	893
weighted avg	0.99	0.98	0.98	893

Figure 8: Classification report

4 Summary and Conclusions

We reviewed the various techniques used to tackle the intention classification task and their successive advances and differences. We achieved approximately SOTA results for the target classification task on the ATIS dataset. This result was achieved by training on a very small dataset of less than 5000 samples. We studied different approaches to the same task and established BERT to perform better than them. Thanks to the almighty BERT, We were able to use a pre-trained model, add a classification layer and get results with percent accuracy. We find that our model is almost always able to generate the correct target. It didn't take long to train such a small dataset, and many of the intricate details of the neural network were abstracted away, saving a lot of time. Although BERT has been able to break many records in downstream tasks, there are other extensions of the BERT model such as RoBERTa and DistilBERT that have surpassed BERT.

Finally, we created a program for the user signing using the system camera and analyse the footage using ASL gesture recognition and intent classification to determine the sign's meaning. The user's communication partner would then see the interpreted message displayed by the output of our program as text.

We used and categorised a sizable collection of ASL gestures and their corresponding meanings using the labels obtained from the intent classification in order to create such a program. To correctly identify and categorise the signs, we trained the deep learning model using this dataset.