Project Report

Natural Language Inference (RepEval 2017)

Submitted by:

Utpal Kumar Dey

UNT ID: 11225732

Introduction: RepEval 2017 shared task [1] was focused on natural language inference. The prime goal was to build model which can transform sentences into fixed-length vector representations and reason using those representations. So, the task was to evaluate natural language understanding based on classified problem over sentence pairs. There are three classes of sentences in the dataset which are: Neutral, Contradiction and Entailment.

Dataset: The shared task features a new, dedicated dataset that spans several genres of text. The dataset is Stanford Natural Language Inference (SNLI) style corpus [2]. The shared task includes two evaluations, a standard in-domain (matched) evaluation in which the training and test data are drawn from the same sources, and a cross-domain (mismatched) evaluation in which the training and test data differ substantially.

Dataset has 393k sentence pairs in training set, 9815 sentence pairs in matched testing set and 9843 sentence pairs in mismatched testing set.

Baseline: Long Short-Term Memory (LSTM) is used as baseline model.

Description of the model is displayed in the following table.

Properties	Value
LSTM	size = 128
Dropout rate	0.2
Recurrent dropout rate	0.2
Activation function	sigmoid
Loss	binary_crossentropy
Optimizer	adam
Metrics	Accuracy
Batch size	128

Number of epochs	10
Maximum sequence length	120

Dataset used:
Training instances = 50000 sentence pairs (40000 for training and 10000 for validation)
Testing instances = 9815 sentence pairs from matched and 9843 sentence pairs from mismatched data set.

Dataset Processing: First, sentence pairs are concatenated to make one line. Then, all the sentence pairs are converted to sequence. After that, those sequences are converted into 2D array, padded with maximum sequence length.

Result:
Matched Accuracy = 68.8%
Mismatched Accuracy = 68.9%

Multilayer Perceptron: In this model, one hidden layer is used in between input and output layer.

Description of the model is displayed in the following table.

Input Layer	
Properties	**Value**
Input dimension	1000
Density	512
Activation function	relu

Hidden Layer	
Properties	**Value**
Input dimension	784
Density	64
Activation function	softmax

Output Layer	
Properties	**Value**
Dropout rate	0.5
Density	4
Activation function	softmax

Other Properties	
Properties	**Value**
Loss	binary_crossentropy
Optimizer	adam
Metrics	accuracy

Dataset used:
Training instances = 50000 sentence pairs (40000 for training and 10000 for validation)
Testing instances = 9815 sentence pairs from matched and 9843 sentence pairs from mismatched data set.

Dataset Processing: First, sentence pairs are concatenated to make one line. Then, all the sentence pairs are converted to sequence. After that, those sequences are encoded using one-hot encoding method.

Result:
Matched Accuracy = 71.8%
Mismatched Accuracy = 72.0%

Character Level Encoding Decoding: This model implements a basic character-level mapping of sequence to sequence. For each sentence pair the premises are being translated character-by-character.

Model Description: Model starts with input sequences from one premise and corresponding target sequences from the other premise. An encoder LSTM converts input sequences to 2 state vectors. A decoder LSTM is trained to turn the target sequences into the same sequence but offset by one timestep in the future.

Dataset used: This model consumes a lot of time to run. So, different chunks of data used for training and testing. The variations of dataset are displayed in the following table.

Training Set			
Total Sentence Pairs	**Neutral Sentence Pairs**	**Contradiction Sentence Pairs**	**Entailment Sentence Pairs**
2100	700	700	700
2700	900	900	900
3300	1100	1100	1100
3900	1300	1300	1300
4500	1500	1500	1500

Testing Set			
Total Sentence Pairs	**Neutral Sentence Pairs**	**Contradiction Sentence Pairs**	**Entailment Sentence Pairs**
300	100	100	100
900	300	300	300
1500	500	500	500
2100	700	700	700
2700	900	900	900

Result:

Training Sentence Pair	**Matched Sentence Pair**	**Mismatched Sentence Pair**	**Number of Epochs**	**Matched Accuracy**	**Mismatched Accuracy**
2100	300	300	5	0.0	0.0
			10	2.2	1.6
			15	2.6	2.1
			20	2.9	2.2
2700	900	900	5	2.1	2.5
			10	2.7	2.6
			15	3.1	2.9
			20	3.2	2.9
3300	1500	1500	5	2.0	2.1
			10	2.5	2.3
			15	2.8	2.7
			20	3.0	3.1
3900	2100	2100	5	0.0	1.0
			10	2.2	1.7
			15	2.9	2.1
			20	3.2	2.5
4500	2700	2700	5	1.2	1.5
			10	2.1	1.8
			15	2.8	2.5
			20	3.4	3.1

From the table above it is obvious that, accuracy increases for more data and more epochs. Even for this small portion of data the model takes much time. So, there is a possibility for better result, if more data are used for more number of epochs.

References:

[1] https://repeval2017.github.io/shared/

[2] http://www.nyu.edu/projects/bowman/multinli/