CS 5984 Assignment 2. Relation Classification

Introduction

Relation classification is one of the pivotal Natural language processing tasks which aims to classify the relation between two entities in a sentence.

The goal of this project is to implement a neural network based architecture for the relation classification task. Given the entity mentions in a corpus of sentences we have to predict what is the relation between the entities e1 - > e2. Additionally we sub sample the dataset by negative samples i.e for an entity tuple e1->e2, if e2->e1 does not exist, we create a new training sample for the entity mentions and annotate them as 'None'(No relation).

Dataset

We use the NYT29 Dataset for the relation classification task. It has 29 relation types (1 extra relation type added annotated as 'None' Type between the entity pair that don't have a relation among them). Each of the datasets viz. Training, test and validation sets consist of .sent files containing all the source sentences and a .tup file indicating the relation tuples from each of the source sentences. Our task is to assume all the entity mentions are given and classify each pair of entities of a sentence to a particular relation type or other (indicating no relation between the two entity mentions).

Choice of Model

In this study I have developed upon the work of Wu et al 'Enriching Pre-trained Language Model with Entity Information for Relation Classification' where they propose the RBert model that both leverages the pretrained BERT language model and incorporates information from the target entities to tackle the relation classification task. The researchers locate the target entities and transfer the information through the pre-trained architecture and incorporate the corresponding encoding of the two entities.

Methodology

1. For the relation classification task we insert special characters between the entity couple to mark them in order to retrieve their contextual word representation retrieved by applying bert over the entire sentences from the training corpus.

2. After retrieving the the word embeddings of the entity mentions, we use their representations as well as the sentence encoding, denoted by the special '[CLS]' token as input to a multilayer neural network for classification.
3. In this way the model captures both the similarity between the meaning of the sentence and fits to find the relation between the two entity mentions by leveraging their contextual representation.

Data Preprocessing.

Preprocessing Steps:

1. We read input sentences from the .sent files and store them into a dataframe.
2. Next for every sentence in the training corpus we extract the entity pairs start index and the end index from the 'train.pointer' file and create relation tuples for every relation present in the files split by ' | ' symbol.
3. Similarly we create entity tuples from the .tup file in which the entity mentions are already given for us to focus on the classification task.
4. Next up we add separators or special characters between entity mentions to process for bert input. For the first entity mention we encapsulate the entity around '$' token and for the second entity mention we encapsulate around '#' token.
5. Next we create negative samples for the entities permutations that are not present in the dataset.

After preprocessing the input dataframe looks like:

	Sentences	relation_pointers	relation_tuples	preprocessed_sentences_labels
0	then terrorism struck again , this time in the...	[{'entity1_start_index': '12', 'entity1_end_in...	[{'entity1_word': 'jakarta', 'entity2_word': '...	{'preprocessed_sentences': ['[CLS] then terror...
1	a12 new york\/region b1-7 enclave for middle c...	[{'entity1_start_index': '16', 'entity1_end_in...	[{'entity1_word': 'stuyvesant town', 'entity2_...	{'preprocessed_sentences': ['[CLS] a12 new yor...
2	before long , though , he 's continent-hopping...	[{'entity1_start_index': '29', 'entity1_end_in...	[{'entity1_word': 'spain', 'entity2_word': 'pa...	{'preprocessed_sentences': ['[CLS] before long...
3	general casey said the iraqi forces had little...	[{'entity1_start_index': '15', 'entity1_end_in...	[{'entity1_word': 'syria', 'entity2_word': 'eu...	{'preprocessed_sentences': ['[CLS] general cas...
4	84 , of avon , connecticut and longboat key , ...	[{'entity1_start_index': '10', 'entity1_end_in...	[{'entity1_word': 'florida', 'entity2_word': '...	{'preprocessed_sentences': ['[CLS] 84 , of avo...

Best Model

After training the model, it achieved a performance of 44% F1 score on the data with a threshold of 0.2. The max_sequence_lengthw as kept around128 which was selected intuitively. The batch size for training was kept around 64 for leveraging the power of GPUs for training. Optimizer used was Adam with a learning rate of 2e-5 and the model was trained for 8 epochs.

micro avg	0.34	0.65	0.44	518
macro avg	0.02	0.07	0.03	518
weighted avg	0.27	0.65	0.37	518
samples avg	0.34	0.67	0.45	518

References.

1. Wu, Shanchan, and Yifan He. "Enriching pre-trained language model with entity information for relation classification." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.
2. NAYAK, T., AND NG, H. T. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In Proceedings of the AAAI Conference on Artificial Intelligence (2020), vol. 34, pp. 8528–8535.
3. RIEDEL, S., YAO, L., AND MCCALLUM, A. Modeling relations and their mentions without labeled text. In Joint European Conference on Machine Learning an Knowledge Discovery in Databases (2010), Springer, pp. 148–163.
4. TAKANOBU, R., ZHANG, T., LIU, J., AND HUANG, M. A hierarchical framework for relation extraction with reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence (2019), vol. 33, pp. 7072–7079.