SBU QA system using tranfer learning

Krishna Sharma
Stony Brook University, NY
skrishna@cs.stonybrook.edu

Hiren Mayani
Stony Brook University, NY
hmayani@cs.stonybrook.edu

Abstract

Due to high volume of data on the internet it is becoming increasingly difficult to search for relevant information. Search engines can be useful to find such information but sometimes it is difficult to answer question asked in natural language. There are several systems available which are trained using large Question-Answer training dataset. These system perform great as they've the data to train their deep neural network but if the system is specific to some domain where amount of data is less, then we will not have enough data to train such complex network and a less-complex network will perform poorly. One such example can be Question-Answering system for Stony Brook University. To develop QA system for such a scenario, We propose methods to learn weights from other large datasets and then fine-tune it using Stony Brook University website data.

1 Introduction

Our objective for this project is to build Question-Answering system for Stony Brook University. We are using transfer learning to train this Question Answering System. Incoming students have a lot of queries regarding different classes, administrative procedures etc which might be present on SBU pages but they've to hunt for them and spend time looking for it. The intention behind developing a Q/A system is to tackle this problem by designing a low latency Question Answering system to help anyone who has questions regarding SBU. This problem is not trivial, companies like Google, AliBaba and IBM are actively participating in competitions like SQuAD leaderboard and publishing research papers regularly.

The basic idea behind Transfer Learning is to pre-train our model on large generic dataset and then fine tune those weights on application specific small dataset. We've gone through several Transfer Learning techniques and research papers and here are the few broad approaches we came across -
1. Unidirectional Models
These models implement standard language model with context window taking into account previous words only and try to maximize the probability of a word given its context words.
$\sum_{i=1}^{n} logP(w_i/w_{i-k},..,w_{i-1};\theta)$
This is a standard technique in Natural language processing used by several recent research papers like Radford et al(2018)[9].
2. Bidirectional Models
Bi-directional models take both past and future context into account while training the neural network. Several models and research papers have used this technique for Transfer Learning like ELMo (Embedding Language Model) [6], BERT(Bidirectional Encoder Representations from Transformers)[7], BiDAF(Bi Directional Attention Flow)[3]. Although some argue that ELMo is shallowly BiDirectional while BERT(Bidirectional Encoder Representations from Transformers) is deeply BiDirectional. The following equation represents the idea of Bidirectional context.
$\sum_{i=1}^{n}(logP(w_i/w_{i-k},..,w_{i-1};\theta_1)$
$+logP(w_i/w_{i+1},..,w_{i+k};\theta_2))$

Although neural networks have been employed before training Q/A systems but using these networks directly for small datasets is a gap that we experienced. Transfer learning bridges this gap but there is no standard procedure or libraries available for transfer learning and is still under research and development which causes developers to face complex setup issues while trying to reproduce results from the research papers. We also explored state-of-art Q/A system to get pre-train weights for our model from the leaderboard on https://rajpurkar.github.io/SQuAD-explorer/. But most of these competitors did not reveal their method and pre-trained model.

BiDAF (Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi) does provide training code. So, we are training it to get pre-trained weights, these pre-trained weights will help to identify structural and semantic information and fine-tuning it will help to answer context specific questions. Also, few days back, Google open-sourced it's top scoring model on SQUAD leaderboard, the Bidirectional Encoder Representations from Transformers (BERT) model. The EM and F-1 score of BERT is far better than any other Q/A system on SQuAD leaderboard. So, we are also trying to use BERT weights for pre-trainied model.

We will use manually written test cases to evaluate our model. We wrote 50 questions and answers related to Stony Brook University. We will use ExactMatch (EM), F1 scores and isSubset to evaluate our model. In ExactMatch, we will count how many answers are exact match of correct answer. In isSubset, We will calculate whether the correct answer is subset of predicted answer or not.

2 Current Progress

So far we were using BiDAF (Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi) to get pre-trained weights. ExactMatch and F-1 score of this was around 60 percent in SQuAD leaderboard. Here is a short summary of the BiDAF research paper.

Bi Directional Attention Flow is a technique used for Machine Comprehension but has been employed successfully in Transfer Learning too. BiDAF used 3 layers for word representation in an effort to capture it at various granularities. These 3 layers are -
(a) Character Embedding Layer
(b) Word Embedding Layer
(c) Contextual Embedding Layer

The fourth Layer is the attention layer which is responsible for creating an attention vector which captures the relationship between context and query. It takes as input context vectors and query vector. Since the attention vector is calculated in both directions - from context to query and from query to context, the model is called Bidirection attention flow. This approach has out perfomed several other approaches like Logistic Regression(baseline), LSTM, Dynamic Coattention networks, R-Net etc.

Preparing data for this system was a big task. As mentioned earlier we wrote 50 test-cases manually. We also wrote script to download text data related to Stony Brook University from stonybrook.edu and wikipedia. We pre-processed this data such that it can be used to fine-tune our model. We installed dependencies and setup development environment for BiDAF. We started training this model 10 days back and it's not completed yet. Currently we are working on fine-tuning code. Once we have pre-trained weights we will fine- tune it using data set we developed.

For BERT, We have set up baseline system and executed it and we are able to re-produce the word embeddings. But we are yet to figure out how can we use these embeddings as pre-trained weights in our QA system. If we are able to do it then we will use it instead of BiDEF.

3 Expected Results

QA system Performance			
Model	ExactMatch	F-1 Score	isSubset
with pre-trained weights Without Fine Tuning	xx.xx	xx.xx	xx.xx
Without using pre-trained weights	xx.xx	xx.xx	xx.xx
pre-trained and fine-tuned	xx.xx	xx.xx	xx.xx

4 Questions we have

As mentioned earlier, training BiDAF is taking really long time. BERT also need Google Credit Platform credit access pre-trained weights. It would be really helpful if it's possible to get some GCP credit for processing.

Other thing which we are yet to figure out is how can we use BERT in QA system given the BERT would just output high quality word embeddings. We are still working on it. If we don't figure it out then we will ask more specific queries during office-hours.

References

[1] https://arxiv.org/pdf/1702.02171.pdf

[2] https://arxiv.org/pdf/1810.04805.pdf

[3] https://github.com/allenai/bi-att-flow

[4] https://rajpurkar.github.io/SQuAD-explorer/

[5] https://www.stonybrook.edu

[6] https://arxiv.org/pdf/1802.05365.pdf

[7] https://arxiv.org/pdf/1810.04805.pdf

[8] https://arxiv.org/pdf/1806.05662.pdf

[9] https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf