International Institute of Information technology – Hyderabad

CSE471 - Statistical Methods in A.I

Course Project report

Project ID :11 Predict movie tagline from text synopsis

Project Guide: Satyam Mittal (satyam.mittal@students.iiit.ac.in)

Project Members (PGSSP Candidates):

1) Jyothsna – 2018900063 (jyothsna.munipalle@students.iiit.ac.in)
2) Surekha – 2018900085 (surekha.medapati@students.iiit.ac.in)
3) Madan – 2018900075 (madan.nandiwada@students.iiit.ac.in)
4) Karthikeyan – 2018900074 (Karthikeyan.arumugam@students.iiit.ac.in)

Table of Contents

Abstract

The problem statement given to us is, to predict the tagline of a movie from given text synopsis. In Other words, this problem statement is to create a Abstract summary (Movie Tagline) from a Document (Movie Synopsis in this case). Hence this is Natural language problem which could use A.I and Machine learning techniques to create a solution. This entire Project work is directed with the technicalities mentioned reference papers [1], [2] and [3]. Our entire solution is implemented using python as programming language. Along with other python tool kits such NLTK, Pandas, NumPy and TensorFlow etc.

From All literature review and tutorials which we found in internet, we arrived to a conclusion to use RNN based model for this problem. Further studies indicated that Seq to Seq Method (also known as Encoder-Decoder) RNN networks are the most suitable model for NLP problems which requires abstractive summary generation.

The Data required to Train our proposed model is List of movie synopsis along with their respective Movie taglines. We used Data listed in Kaggle Dataset (Mentioned in Ref [7]). After Clean up of the raw data we used Concept Net's Number batch-based pre-processing to create word embedding matrix.

The Final Model Developed during course of this project was able to create a movie tag line. This model is created to handle Movie synopsis and movie tag line documented in the English Language only.

Keywords : NLP, TensorFlow , Recurring neural networks, Sequence to Sequence , Number Batch, Python

Tools and Technology Used

- *Programming language: Python 3.X*
- *Dev tools: jupyter notebook notebooks, Colab.research.google.com*
- *Toolkits Used: TensorFlow, NumPy, Pandas, etc.*
- *SCM: Git Hub and Google Drive(for Large files)*
- *Solutions Type: Recurring Neural Network*

Quick Reference Links :

GIT Hub Repository URL: https://github.com/karumugamio/SMAIProject-Team31

Google Drive URL: http://tiny.cc/team31gdriverepository

Challenges

Data

- Missing values
  The Data set from Kaggle contains over 45000 movie details. Unfortunately, there are lot of data points where either synopsis or tagline. This renders those data points use less as we couldn't use them to train the model.
- Non UTF8 Text contents
  A lot of international movie details contains non UTF 8 Characters. These unknown chars may include trademark /copyright symbols, Latin chars. Synopsis and tagline Text which are typed in a language which not in English.
- Non-English Text Contents

  There are many data points in the Kaggle data set which contains non-English characters. Such as Movie synopsis or tagline written in regional Language fonts which are a valid English text content. *Example: Tagline given in Telegu or Malayalam*.

Unknown words in Data set

There are many out of vocabulary words in out data set. Such unknown words tends to not contributing to our training weights There is always possibility of getting a Unknown word which is not part of word embedding matrix which we created. i.e. Out of Vocabulary Words. Such out of vocabulary will be assigned as unknown word in embedding it will not help model to train better.

Examples: Name of Characters in the movie like Sivagami, Baahubali etc

Challenges with Training model

a) Huge Volume of data to be processed

   Each Plot Synopsis is a para or list of sentences each has 10-15 words per sentences.

   When we convert all words in the given synopsis into word vectors. Processing entire data set with each 3-4 sentences per synopsis will leads to Spike in the Volume of data handled by the system

b) Computation complexity and Long hours of training

   As we supply a huge set of movie synopsis and plot lines into model, each one of them these data points have their own complexity with data. Mathematical operations over a large matrices takes toll in system memory. Hence for many parts of the solution we should try it in google colab with GPU enabled option.

c) Risk of Losing training data while executing training for long hours

   As per are planning to process data as batches , after each batch computation completion there is a challenge with holding weights in memory of the program. Any interruptions and

errors may lose entire training data. Hence Holding weights in memory during computation is risky.

Data analysis and Clean up

- *Data source:* Kaggle Link https://www.kaggle.com/rounakbanik/the-movies-dataset
- Data source contains many features which are not required by our model, he removed all features except synopsis and tagline.
- Reduced Data frame is then checked and cleaned up.
- Clean up also includes filtering out non UTF8 characters and Non-English Characters in the Data.
- Train, validation split is not done in data clean up stage, to ensure we capture all possible words in Word embedding matrix. Hence during validation, system will not categories common words in data set as known.

Solutions and workarounds

1. Training in batch

   In order to handle computational complexities, we train the RNN in batches. Each batch size is left as configurable parameter

2. Check points for each batch

   While training our model using tensor flow we can save the model weights as a checkpoint file. This Check point file can be appended with values in sub sequent batch of training.

   After Completing our training we can distribute the trained model using this checkpoint meta data files. Any user can load the model weights and use it predict without actually running through long training hours.

Implementation

The Model we implemented for this RNN project contains Bi-Directional LSTM.

RNN consists of two sub models, first part of the RNN is called as Encoder and the output the first sub model is fed as input to the second sub model (Decoder).

We use Adam Optimizer in our training model.

Entire Data fed into Training is spliced into N number of Batches.

At end of each Batches we do update the Training weights with help of tensor flow built in methods to save it as check points

Stop Criteria, In spite of no of Epochs in the training, model looks out the change in loss values in each batch. In any given epoch if the loss value remains same for three consecutive iterations we will stop training the model further.

When out learning rates adjust according to learning decay rate and if it falls below threshold value which we set the training will stop.

Once model is trained , the weights saved during check points as meta data can used to any time to load this model back and perform prediction.

Performance metrics

As per our literature review we found various performance score which can be used to compare two sentences. For example , Cosine Similarity , Rogue Score and BLU score etc.

Practical challenge with this problem statement is , the movie taglines for the movies are human labelled data. Usually human labels are arrived contextually with creativity. Also most of the times tagline given by artist is arrived implicitly by various other factor.

Example : The movie Baahubali – Part 1 , have Tagline as " The Beginning". There is no specific mention of this in given text synopsis. Hence Tagline predicted from Synopsis may not have common words between tagline artistically created.

Rogue score , Cosine similarity etc works purely based on the common words, hence measuring performance with any of this metrics most of the time results in zero.

Our code changes to measure the same resulted in zero for most of the Validation data points. Hence we didn't measured performance metrics for this model.

Possible Future Work

This solution can be extended to the other language in future. In particularly to use Regional Indian Language text as Input Synopsis and to generate tagline in their respective language requires Language corpus and Knowledge tree map similar to Number Batch for those languages. Unfortunately, ConceptNet Number batch don't have corpus for Indian Languages yet. Recreating this model for regional Indian language heavily depend upon local language corpus.

Conclusion

With Help of RNN model we where able to train a model with English language. Trained model was able to generate tag line for given Movie Synopsis. The Final Code library is available in GIT link mentioned in the beginning of the document. In order to deploy this model in any other environments we have saved training weights and data as TensorFlow checkpoints. The Checkpoints and other data are saved in the Google drive folder due to file size restriction in GitHub.

References

1. Predicting Movie Genres Based on Plot Summaries by Quan Hoang : https://arxiv.org/pdf/1801.04813.pdf
2. Folksonomication: Predicting Tags for Movies from Plot Synopses UsingEmotion Flow Encoded Neural Network by Sudipta Kar, Suraj Maharjan and Thamar Solorio - https://aclweb.org/anthology/C18-1244
3. Patent Abstract Summarization using Recurrent Neural Networks by Abhishek Jindal,Chirag Choudhary and Nile Hanov – https://github.com/ajindal1/Text_Summarizer_On_Patents/blob/master/project_report/Text_Summarization_project_NLP.pdf
4. Rouge Score: https://stats.stackexchange.com/questions/301626/interpreting-rouge-scores
5. Performance metrics - https://nlpprogress.com/english/summarization.html
6. RNN – Sequence to Sequence Model : https://towardsdatascience.com/seq2seq-model-in-tensorflow-ec0c557e560f
7. Data Source:
    a. Found in Kaggle - https://www.kaggle.com/rounakbanik/the-movies-dataset
    b. Filename : movies_metadata.csv
8. Concept Net Number batch – https://github.com/commonsense/conceptnet-numberbatch
9. Glove - https://nlp.stanford.edu/projects/glove/