Digest generation for the news articles using LSTMs.

Mikhail Sidorov
CS230
msidorov@stanford.edu
Category: NLP

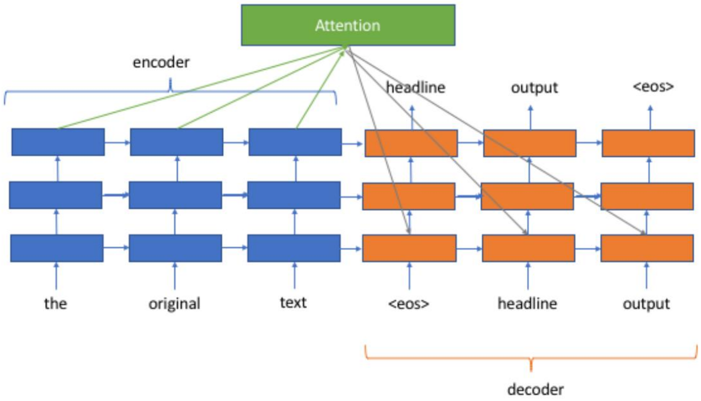
Abstract

The problem of representing document in short form is very important in different areas including news aggregation and providing search results. Last years text summarization algorithms have been significantly improved due to applying of novel deep learning techniques. In our project we would like to try and compare the ideas proposed in latest articles [1,2,3]. The scope of the project includes applying various RNN-LSTM- based summarization approaches to the text and prepare summaries for text documents.

1 Introduction

Text summarization - creating a short, accurate summary which contains the main information from the original text, continues to be inetersing for research machine learning topic. The theory has a great potential to be used in various applications, including: web and product search (both retreiving and representations of the results), media monitoring, newsletters, social media marketing, question answering and bots, medical cases, books and literarture, automated context creation.

The focus of our work is in applying and comparing different abstractive single-document abstractive summarization methods. Important aspect of the project is that implementation should handle texts with out-of-vocabulary (OOV) words. For text summarization we use sequence-to-sequence model with attention mechanism.



Submitted as a project milestone for CS230. Winter, 2019.

The loss function in this case is:

$$-log\,p(y_1,...,y_{T'}|x_1,...,x_T) = -\sum_{t=1}^{T'} log\,p(y_t|y_1,...,y_{t-1},x_1,...,x_T)$$

For word embedding representation we used GloVe.

The result of our work is a trained model in tensorflow which is able to take as input any news and generate a summary. Also we use the network to other domain (eCommerce, to generate annotation of food review based on the review text).

2 Related work

There are two main approaches to automatic text summarization: extractive and abstractive. Extractive approaches select passages from the source text, then arrange them to form a summary while abstractive approaches use natural language generation techniques to write new sentences [8].

The seminal work [5] in this area suggested to apply Sequence-to-sequence learning using encoder-decoder implementation based on LSTM/RNN with attention mechanism demonstrated impressive results for different areeas including machine translation, text summarization etc. Several metrics (BLEU and it's derivatives: METEOR and ROUGE were suggested to mesure the quality of the output).

Seq2seq NLP get a benefit of using word embedding which helps to grasp the semantic of text using either pretraing (like GloVe) embeddings or by including training embeddings as part of seq2seq model training.

The main issues known for abstract text summarization implementations are [1]:

1) Inaccurate reproduction of factual details

2) Summaries repeat themselves

Different techniques are used to overcome these issues and improve the quality of text summarization.

Our project focused on applying Deep Learning NLP [9] for abstracive text summarization and we investigated several methods mentioned in [1] and used tensorflow and keras for the implementation.

3 Dataset

There are several well-known avaliable datasets which we can use for our project. Most of datasets mentioned in atricles [1,2,3] are publicaly available (except Stanford Gigaword dataset). Very good observation of available datasets is provided in [4].

For training and verification of the approaches we will use Cornell Newsroom dataset (`https://summari.es`) which has been recently revealed (May 2018). The summaries are obtained from search and social metadata between 1998 and 2017 and use a variety of summarization strategies combining extraction and abstraction.

Table 1: Newsroom dataset statistics

Dataset size (articles)	Mean article length (words)	Mean summary length (words)	Total vocabulary size (words)	Occurring 10+ times (words)
1,321,995	658.6	26.7	6,925,712	784,884

The example of content of Newsroom summarization dataset is provided below:

SUMMARY: A young woman has been arrested after allegedly glassing another woman during a wild brawl at a Sydney train station.

ARTICLE: A wild brawl between two women at a Sydney train station has left one with head injuries after she was struck with a glass bottle. The women were fighting at Redfern train station just

before 1am on Friday before police used pepper spray to break
them up. One of the women, aged 27, was taken to hospital after
copping a bottle to her head. The other woman, aged 20, was
arrested , questioned and later released as investigations
continue . ...

Also we developed our method and applied text summarization technique for Amazon food reviews. This data set is provided on kaggle site: https://www.kaggle.com/snap/amazon-fine-food-reviews

This data set is significantly smaller (about 586K examples), with short reviews and short summaries and first we trained our neural network on this dataset (used different hyperparameters for these 2 datasets).

Table 2: Food review dataset statistics

Dataset size (articles)	Mean article length (words)	Mean summary length (words)	Total vocabulary size (words)	Occurring 10+ times (words)
568,454	80	4.1	132,884	64,183

The typycal example is:

SUMMARY: Great Irish oatmeal for those in a hurry!

ARTICLE: hurry! Instant oatmeal can become soggy the minute the water hits the bowl. McCann's regular oat meal...

The main motivation to use the second dataset was to debug the tensorflow program on more simple dataset and observe the model performance issues. More simple language structure of the second data set allows neural network with the same complexity to achive better performance.

3.1 Preprocessing

Summaries and texts were lowercased and tokenized. Also we used a list of english langguage contractions for preprocessing to convert them in regular words. An <EOS> has added to text. We preserved number of words using a threshold so that we didn't exceed 100.000 words which were most often used words. Unkown words were replaced with <UNK> symbol.

The data has been splited into a training/dev/test sets. Training set has been randomly reshuffled. Dev set has been used for hyperparameter tunning and Test set has been used for final estimation of the summarization quality.

4 Metrics

During the training nof the model we optimize the loss function, but to evaluate model preduction we use discrete metrics (whichbecause of this is not be able to be used as a loss/target function).

We will follow atricles [1,2,3] and evaluate the results of summarization using metrics: Recall-Oriented Understudy for Gisting Evaluation (ROUGE-1, ROUGE-2, ROUGE-L). According [6] ROUGE-N which is N-gram co-occurence statistics which is calculated as n-gram recall between a candidate summary and a set of reference summaries.

$$ROUGE-N = \frac{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count(gram_n)} \quad (1)$$

and $ROUGE-L$ is defined as Longest Common Subsequence between candidate and reference summaries.

In refered atricles Metric for Evaluation of Translation with Explicit ORdering (METEOR) also is used.

Metrics for the Newsroom dataset [4] represented in table 2.

Table 3: Current F-score ROUGE metrics for Newsroom dataset

Method	R-1	R-2	R-L
Lede-3 Baseline (2018)	30.63	21.41	28.57
Pointer generator (2017)	26.04	13.24	22.45
TextRank (2016)	22.76	9.80	18.97
Seq2Seq + Attention (2015)	5.91	0.43	5.36

5 The results

First we developed a software with enough flexibility to quickly verify and estimate the performance for different sets of the hyperparameters. Than we estimated the performance of the model on the Amazon food review dataset and tune network architecture as well as GloVe dimension and set.

The structure of software included:

1) Setup code: responsible for downloading data from AWS and upload model to AWS

2) Data load module - code responcible for loading data (train/dev/test)

3) Text parsing and cleanup

4) Vectorization of tokens (our implementation is based on GloVe)

5) Model definition

6) Prediction and performance estimation

We collected the following results:

Table 4: ROUGE metrics for Amazon food reviews dataset

Method	R-1	R-2	R-L
F	15	3	13
Precision	20	5	20
Recall	14	3	14

We made tunning for the NEWSROOM datset as well and got different set of optimal parameters (4 hidden layers and GloVe)

Table 5: ROUGE metrics for NEWSROOM dataset

Method	R-1	R-2	R-L
F	5	0.5	4
Precision	6	0.3	6
Recall	4	0.2	4

We found that for short summaries scores where better that for small summaries for the Food data set caused by more simple language structure.

6 Code repository

While our initial implementation has been done in keras: https://github.com/masidorov/cs230.train/blob/master/Seq2SeqV1.ipynb the final implementation has been done in tensorflow

We provided a complete solution - the python script with set of parameters (see main.py file) which provides an opportunity to download dataset from AWS and traing the neural netwoord with different parameters, including:

1) The GloVe file and dimensionality of the word embedding

2) hyperparameters of the neural network and training including the learning rate and number of layers

3) The dataset (either NEWSROOM dataset or Amazon food review dataset)

We used parameterization to train the model with different hyperparameters to achive the best results. In each case we used train, dev and test datasets. We used dev sets to get the optimum hyperparameters and verify the final performance on test dataset.

Current code with implementation of the seq2seq network has been provided here:

The final project repository:https://github.com/masidorov/CS230.ZX/tree/master

The code itself has been arranged in such way, that we trained the model on remote computers in AWS cloud, but downloaded the model on MacBook and were able to make evaluation and use the saved model to make summarization using the notebook.

7 Conclusion

In presented project we trained an encoder-decoder neural network with LSTM units and attention for text summarization problem. We used 2 completely different data sets and got much better result on data set with short summaries and much more simple language structure, which we can expect a priory. Approach demonstrated the fesibility and definitely it will be interesting to apply for this problem the latest techniques which were probposed in the latest articles.

References

[1] Abigail See, Peter J. Liu, Christopher D. Manning (2017)Get To The Point: Summarization with Pointer-Generator Networks. 2017, ACL. (`https://arxiv.org/pdf/1704.04368.pdf`)

[2] Alexander M. Rush, Sumit Chopra, Sumit Chopra, Jason Weston (2015). A Neural Attention Model for Abstractive Sentence Summarization. `https://arxiv.org/pdf/1509.00685.pdf`

[3] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, Bing Xiang (2016). Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. `https://arxiv.org/abs/1602.06023`

[4] Max Grusky, Mor Naaman, Yoav Artzi (2018) NEWSROOM: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. `http://aclweb.org/anthology/N18-1065`

[5] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. (2014.) Sequence to sequence learning with neural networks. In Neural Information Processing Systems

[6] Chin-Yew Lin, (2004) ROUGE: A Package for Automatic Evaluation of Summaries. Information Sciences Institute, USC

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee Kristina Toutanova (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding `https://arxiv.org/pdf/1810.04805.pdf`

[8] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assef,.. (2017) Text Summarization Techniques: A Brief Survey `https://arxiv.org/pdf/1707.02268.pdf`

[9] Li Deng, Yang Liu, (2018) Deep Learning in Natural Language Processing, Springer.