INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Taste and Tell: An automatic review generator and Restaurant Recommender system

Final Project Report for CS671A: Introduction to Natural Language Processing

Vasu Sharma (12785)
Gundeep Arora(16111010)
Peshal Agarwal(13472)
Project Guide : Prof. Harish Karnick

November 20, 2016

1 PROJECT IDEA

The project is based upon the foremost desire of human existence... Food. In this project we plan to build and automatic review generator and recommender system which is a one stop solution for all foodies out there. We offer a wide array of features which will make the task of going through an enormous amount of reviews and extracting information a lot easier and directly analyze such reviews to provide users with valuable recommendations based on their own food choices and the choices and behaviours of all the other users. We also create an automatic review generator which makes it possible to generate automatically written reviews based on the kind of rating a particular reviewer gave to a restaurant. The generated reviews retain the style of the review writing from a reviewer to make it feel more personalized, an ideal thing for all those lazy food reviewers out there who don't intend to spend their valuable time writing food reviews.

1.1 PROJECT FEATURES

We implement the following functionality in our project:

- Sentiment Analysis from reviews. This includes going through the food reviews and trying to gauge user sentiment and assign a score based on it. Score parameters have been found to be much easier to go through and base ones decisions upon rather than manually going through hundreds of food reviews.

- Automatic Recommender System: We build a recommender system on top of the automatic analysis of the food reviews. Such a recommender system effectively uses the history of the user as well as the history of other similar users to predict what a particular user might be interested in. The additional novelty we implement here is to come up with a Deep Learning Based Recommennder system which outperforms the existing state of the art collaborative filtering approach.

- Finally, we implement an automatic review generator. Producing Natural language text which makes sense has known to be a fairly challenging problem for a computer to learn due to the presence off inherent rules and structure in language which is fairly hard for the computer to learn. We use recurrent neural networks which have been known to show amazing prowess in remembering context and learning such rules given sufficient data. Such a review generator system caters to each individual users reviewing style and would convert a user provided rating into a full fledged review personalized to the users writing style and based on the rating he provided. As far as we know such a system has never been made before with such levels of personalization and hence is a challenging task to perform.

- One of the major challenges with this task is that of the lack of a metric to test the efficiency of the system. Plus the review generator and sentiment analysis system are

standalone systems in most cases which prevents it from collaboratively training each other. We introduce the novel concept of joint training of both systems which not just provides us a metric to test the efficiency of our review generator but also helps to better train both the networks as they can now be trained in an end to end manner and will adapt better to this specific task.

- In addition to the above things which had formed a part of our initial proposal, we also implement a review summarizer which summarizes the reviews to retain only the relevant parts while trimming out the unnecessary parts. This could help the audience by only presenting to them the important parts of a review while ignoring the irrelevant parts.

2 APPROACH

We took this project as an opportunity to explore a variety of existing frameworks and methods and come up with new ones to improve upon them. The main things we tried are as follows:

1. **Recommender System**: We implemented a Collaborative Deep Filtering approach based on Bayesian Stacked Autoencoder networks where the encoder-decoder framework is trained in an unsupervised fashion using a corrupted version of the input vector and the network tries to recreate the original denoised vector. The vectors are drawn from latent bayesian space which is learnt using the user-rating data. The latent distributions from which the vectors are sampled are learnt using EM style algorithms and the estimated parameters are then used to perform the predictions by casting the rating problem as an Expectation calculation problem. It is basically able to form effective deep feature representation from content and capture the similarity and implicit relationship between groups of items and users. The mathematical details of the same are very rigorous and hence we won't go into the same here.
   We implemented this Deep Learning framework and evaluated it on the Amazon Fine food reviews dataset and achieved results which outperformed the collaborative filtering approach. The results are presented in the results section.

2. **Review Summarization**: Review summarization is an effective and handy feature to provide to the users as it allows them to only go over the relevant parts of a review. We implemented Review Summarization using 3 different techniques and evaluate them using ROGUE metric for a different dataset for which we have access to human summaries.(See Results section). The various techniques we use are:
   - LexRank: Graph-based Centrality as Salience in Text Summarization:- It is based on eigenvector centrality of graphical representation of sentences. Cosine similarity between sentences is used as an adjacency matrix. It represents sentence as the N (no. of all possible words) dimensional vector with the value of each dimension equal to frequency of word times the idf of word. The similarity is defined to be the cosine between vectors. Cluster of document is defined as the cosine similarity

matrix. This matrix can also be represented as a weighted graph where each edge shows the cosine similarity between a pair of sentence. Degree centrality of a sentence is defined as the degree of the corresponding node of the similarity graph. A better way is have centrality value of each node and distribute this to its neighbours. The adjacency matrix turns out to be a stochastic matrix. The algorithm starts with a uniform distribution. At each iteration, the eigenvector is updated by multiplying with the transpose of the stochastic matrix. At convergence, it gives us the LexRank vector.

- Using Latent Semantic Analysis (LSA) in Text Summarization and Summary Evaluation: This method deploys SVD for a matrix where each row represents a frequency weighted sentence vector to extract semantic features of each sentence in a latent space. The magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. The summarization algorithm uses the most informative sentence for each topic.
- New Methods in Automatic Extracting (Edmundson) : We try to select sentences from the documents that best convey the meaning of the entire document. It uses four components namely, pragmatic words (cue words); title and heading words; and structural indicators (sentence location), instead of using just key words. The training is performed on manually created extracts and model is improved by comparing with automated ones. It gives positive weights to for desired sentences and penalty weights for undesired sentences.
  - Cue-weights sentences according to match with Cue dictionary.
  - Key-weight according to frequency of words
  - Title-weight according to match of title and heading
  - Location-weight according to location in the document

  For simplicity overall weight is sum of the weights given to each of the four characteristics.

The summarization produces coherent and exhaustive summaries which captures the major essence of each of the reviews. We notice that the LexRank algorithm outperforms the other 2 algorithms as can be seen from the results in the results section.

3. **Sentiment Analysis**: We implement a Sentiment Analysis pipeline which uses Paragraph vectors (Doc2Vec) for generating vector embeddings of reviews.
   Doc2Vec (gensim) in purely unsupervised mode needs no labels other than an arbitrary unique ID per text example. It generalizes Word2Vec to whole documents. At first, fixes the length of vector. Then, assigns and randomly initialize document vector to each document.

   It tries to predict the next word using the paragraph vector and the context words. The context window is being slided keeping the paragraph vector being fixed. Then the document vector is being updated using the stochastic gradient method. Once we have these document embeddings, we feed it to our classification pipeline which assigns

these documents into various categories representing the score alloted to them. Scores are assigned as integers in the range of 1-5. We experiment with a variety of different classification techniques. The results of the same are present in the results section. We also try out the LSTM based sentiment analyser which directly takes the text document as input and keeps updating the LSTM state with each input word. Once the whole document has been seen (terminated by the special symbol $) then it generates a score for the document.

4. **Review generator network**: The review generator network we implement is based on the char-rnn model. It uses an RNN pre-trained on the Google News corpus and we finetune it on our dataset. We modify the network to take in the user id and a score the user wants the review for. This information helps the network personalize the review to each user and also correspond to the rating the user wants the review for. The output of this rnn is a sequence of characters. We keep running the network till we have obtained the desired length of the review.
   The goal of character-level language modeling is to predict the next character in a sequence. More formally, given a training sequence $(x_1, ..., x_T)$, the RNN uses the sequence of its output vectors $(o_1, ..., o_T)$ to obtain a sequence of predictive distributions $P(x_{t+1}|x \leq t) = softmax(o_t)$. The language modeling objective is to maximize the total log probability of the training sequence $\sum_{t=0}^{T-1} logP(x_{t+1}|x \leq t)$, which implies that the RNN learns a probability distribution over sequences. Even though the hidden units are deterministic, we can sample from an MRNN stochastically because the states of its output units define the conditional distribution $P(x_{t+1}|x \leq t)$. We can sample from this conditional distribution to get the next character in a generated string and provide it as the next input to the RNN. This means that the RNN is a directed non-Markov model and, in this respect, it resembles the sequence memoizer. Image 2.1 explains this network.

5. **Joint training model of sentiment analyser and review generator**: We now put together the sentiment analysis network with the review generator network. The output from the review generator network becomes the input to the sentiment analysis network. The Correlation between the sentiment analysis scores of the generated articles and the actual scores fed while generating the articles is used as an evaluation metric for the review generator network. We first train both the networks separately and then jointly finetune both networks in an end to end manner using the euclidean distance between the original and predicted scores as the error metric.

3 Results

3.1 Restaurant Recommender System

The Recommender system trains well and quickly learns to make fairly accurate recommendations. The plots show how the training and test error varies with iterations. This Deep

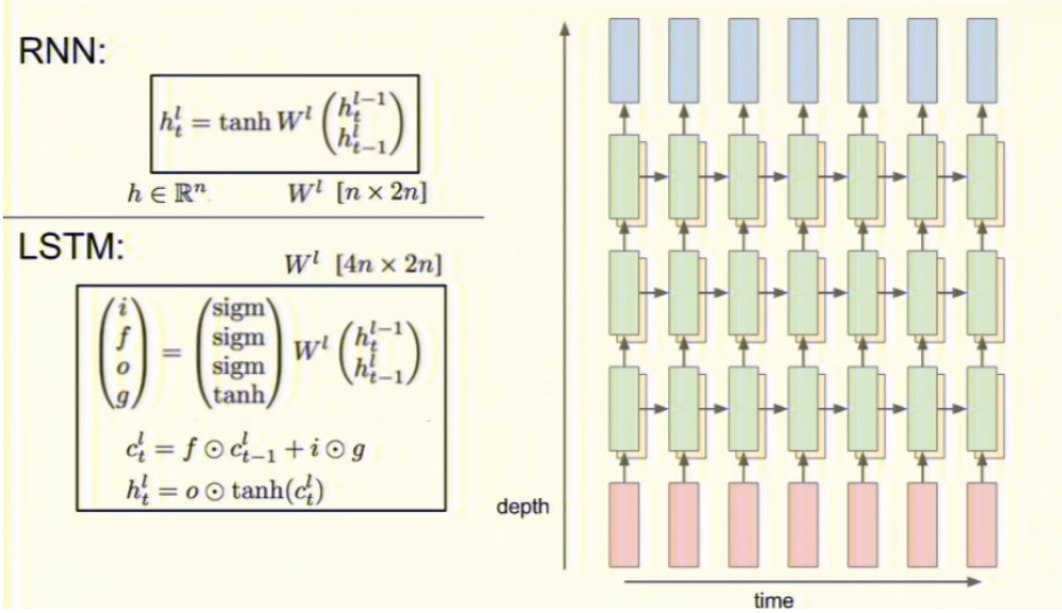


Figure 2.1: Review generator network

Recommender system outperforms the traditional Collaborative Filtering approach by a substantial margin. The table below shows the comparison of the results on both the test set of Fine Food reviews dataset and on the dataset augmented with the Yelp Food reviews dataset.

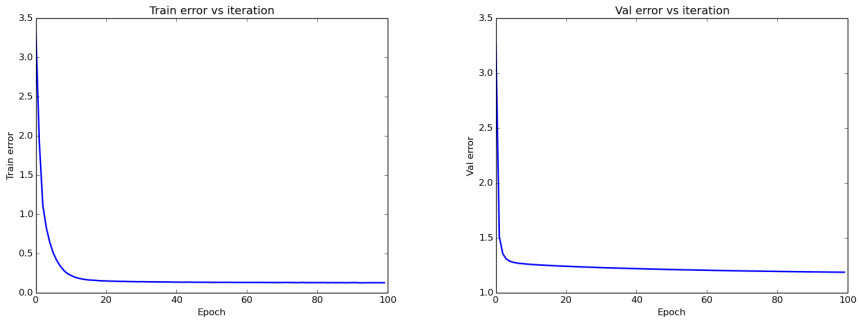
Algorithm	Test set error	Error on augmented dataset
Collaborative Filtering	1.425	1.287
Deep Recommender system	**1.186**	**0.822**

Table 3.1: Mean Squared errors for recommender system

3.2 Review Summarization

As we do not have the ground truth summaries for this particular task hence we will not be able to present the results of our summarization techniques on the given Amazon fine food reviews dataset. Instead, we present a few illustrative examples of our method in action on this dataset and present a more quantitative evaluation on the DUC-2001 Single document summarization task. We implemented 3 document summarization techniques, namely LSA[4], LexRank[2] and Edmundson[1] method for summarization and use the ROGUE(Recall Oriented Understudy for Gisting Evaluation) metric [3] as the evaluation parameter, which compares the automatically generated summaries and the human created summaries by counting the overlapping word counts, n-grams, word sequences and word pairs.

We also present an illustrative example on the Amazon fine food reviews dataset:



Dataset	LSA	Edmundson	LexRank
DUC-2001	0.48	0.42	**0.58**

Table 3.2: Rogue-1 Scores for document summarization for DUC dataset

Actual review:"""My Cats Are Not Fans of the New Food. My cats have been happily eating Felidae Platinum for more than two years. I just got a new bag and the shape of the food is different. They tried the new food when I first put it in their bowls and now the bowls sit full and the kitties will not touch the food. I've noticed similar reviews related to formula changes in the past. Unfortunately, I now need to find a new food that my cats will eat."

Summary generated:"""My cats have been happily eating Felidae Platinum for more than two years.
I've noticed similar reviews related to formula changes in the past.
Unfortunately, I now need to find a new food that my cats will eat.
"""

3.3 Sentiment Analysis

The following table3.3 describes the results for the Sentiment Analysis method:

3.4 Joint training model of sentiment analyser and review generator

The following table 3.4 and 3.5 describes the results for the Joint training:

4 Challenges so far and expected challenges

- Reproducing the results similar to those mentioned in the papers is a common challenge that we faced initially and expect to face in the future.
- With very deep networks it is possible that the network size may be too big to load in memory alongwith batch data during the training.

Feature Representation	Classifier	Accuracy(%)
BoW	Linear SVC	68.2
BoW	Perceptron	69.7
tf-idf weighted BoW	linear SVC	72.5
tf-idf weighted BoW	Perceptron	75.6
BOV	linear SVC	70.2
BOV	Perceptron	72.3
Direct LSTM	N.A.	89.7
Paragraph Vector	Linear SVC	85.5
Paragraph Vector	Perceptron	88.2
Paragraph Vector	**SVM+RBF kernel**	**92.6**

Table 3.3: Accuracy Scores for Sentiment Analysis

Feature Representation	Classifier	Accuracy(%)
Direct LSTM	N.A.	90.6
Paragraph Vector	Linear SVC	87.4
Paragraph Vector	Perceptron	90.3
Paragraph Vector	**SVM+RBF kernel**	**94.8**

Table 3.4: Accuracy Scores for Sentiment Analysis after finetuning

Finetuned(yes/no)	correlation score
No	0.87
Yes	**0.94**

Table 3.5: Accuracy Scores for Review generator after finetuning

- Text generation is an extremely challenging part and with small data available for every user, personalizing the generation of text becomes very difficult. To the best of our knowledge, there is no formally published work around this.
- The dataset is too small for summarization, hence some part of Yelp dataset had to be augmented for training purposes.

REFERENCES

[1] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 1969.

[2] G. Erkan and D. R. Radev. Lexrank: Graph-based centrality as salience in text summarization, 2004.

[3] C. Y. Lin. Looking for a few good metrics: Rogue and it's evaluation.

[4] J. Steinberger and K. Jezek. Using latent semantic analysis in text summarization and summary evaluation. *ISIM*, 2004.