Machine Learning Capstone Project Proposal

Yan Kang

1. **Domain background**

With the rise of online social media platforms such as Twitter, Facebook and Weibo, and the proliferation of customer reviews on e-commercial sites like Amazon and Yelp, it is increasing important that companies are able to accurately analyze customers' sentiment around reviews for products and feedback for advertising campaigns. This sort of analysis is called sentiment analysis and it becomes probably one of the most popular applications of Natural Language Processing these days.

However, most models in sentiment analysis use bag of words (BoW) representations [1] that ignore word orders and thus they cannot accurately address hard sentiment analysis issues, such as negation and sense ambiguity. Recurrent Neural Networks (RNN) [2] is able to address these hard issues to some degree as they treat text as sequences. Recursive Neural Networks [3] takes into account the syntactic structure of text and thus they also are capable of partially resolving these issues. In this project, we develop a RNN model and a Recursive Neural Network model for solving a sentiment analysis problem, and use BoW-based Naïve Bayes model as baseline model to evaluate the performance of the two models.

2. Problem statement

In this project, we will use three different models to perform the sentiment analysis task on movie reviews. More specifically, we use Naïve Bayes model, Recurrent Neural Networks (RNN) and Recursive Neural Networks.

The Naïve Bayes model serves as the benchmark model, based on which the performance of RNN and Recursive Neural Networks are analyzed. Since the Naïve Bayes does not take the positions or orders of words into account, it has difficulty in addressing issues such as negation and sense ambiguity. RNN has been proven to be very successful at addressing such issues because it treats text as a list sequences and incorporate information over time. However, RNN does not fully exploit the syntactic structure of text, while Recursive Neural Networks computes compositional vector representations for phrases of variable length and syntactic type, and empirically, it is proven to perform well on sentiment analysis [3].

We will evaluate and compare the performance of the three models based on the accuracy of classifying customer movie reviews.

3. Datasets

We use dataset from Stanford Sentiment Treebank [4]. This dataset is designed for Recursive Neural Networks:

- Each sample in this dataset is represented as a parse tree that is structured to identify dependencies between words.
- Each node in a parse tree is labeled using Amazon Mechanic Turk [5], and have rating: 0-4, in which 2 is neural, 0 is the most negative and 4 is the most positive. For example, the parse tree of review "A deep and meaningful film." is depicted as follow:

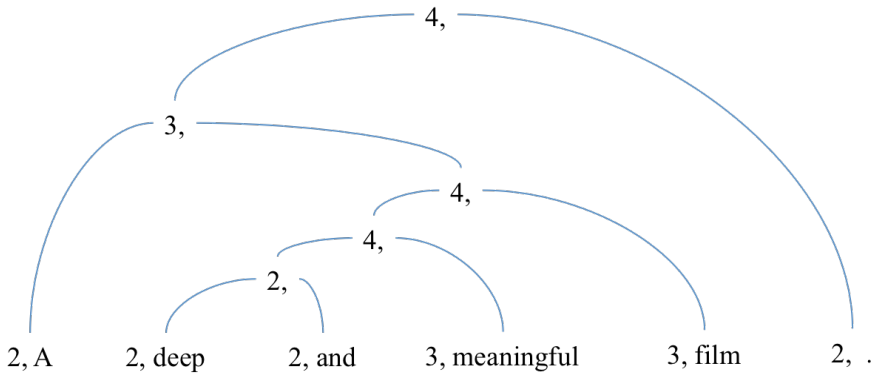

Figure 1: parse tree of sentence "A deep and meaningful film"

In this project, we only perform positive/negative classification. Therefore, we label all samples with rating 0-1 as negative, all samples with rating 3-4 as positive, and ignore samples with rating 2. After transforming, we have 6920 training samples and 1821 test samples. 3310 of the 6920 training samples are negative samples while the rest 3610 are positive samples. Therefore, There is no imbalanced class issue.

We transform all samples in the form of parse trees into bag-of-words vectors and sequences such that we can use Naïve Bayes model and RNN to train these samples.

4. Solution statement

When data have all been properly transformed, we feed them into models. To get a benchmark model, we train a Naive Bayes classifier from scikit-learn [6] (specifically GaussianNB). We utilize TensorFlow [7] framework to design a Long-Short Term Memory RNN and utilize Theano [8] framework to implement a Recursive Neural Networks.

The reason we use TensorFlow for designing RNN is because TensorFlow provides efficient implementation for main components (e.g., LSTM cell) required by a RNN. The only thing left for us to do is to design architecture for the RNN.

The reason we use Theano to implement Recursive Neural Networks is two-fold:

- Theano is a lower level framework such that it provides more flexibility for implementing new models.
- Learn the Recursive Neural Networks in a more thorough way while implementing the algorithm

When all models have been trained, we evaluate and compare the accuracies of the three models based on the test data.

5. Benchmark model

We use built-in Naïve Bayes model with default parameters from scikit-learn as the benchmark model. The Naïve Bayes model can achieve decent performance (around 80% accuracy) on sentiment analysis problem. However its performance cannot be significantly improved since it has difficulty in addressing issues such as negation and sense ambiguity. Using Naïve Bayes model as benchmark is an appropriate choice since it can really verify whether RNN and Recursive Neural Networks have the edge on addressing those difficult issues.

6. Evaluation metrics

In this project, we mainly use accuracy to evaluate the overall performance of models. Accuracy takes into account both true positives and true negatives with equal weight:

$$accuracy = \frac{true\ positives + true\ negatives}{dataset\ size}$$

Because we have no imbalanced class issue (we have 3310 negative samples and 3610 are positive samples, using accuracy is sufficient.

We also pay close attention to the accuracy of classifying negative samples since this is where the RNN and Recursive Neural Networks may outperform Naïve Bayes model.

7. Project design

7.1. Data processing

The raw samples are stored in serialized form of parse threes. We implement algorithms that can transform raw samples into the format that is suitable for each of the three models respectively.

1. Transform each of the raw samples into a tree structure that can be processed by Recursive Neural Networks. Particularly, post-order tree traversal algorithm is used in this transformation.

2. Transform tree structure version of samples into sequences for RNN.
3. Transform tree structure version of samples into bag-of-words vectors for Naïve Bayes Model.

7.2. Model Development

1. Utilize built-in Naïve Bayes model from scikit-learn as the benchmark model
2. Design architecture for the RNN by using Tensoflow
3. Implement the Recursive Neural Networks algorithm by using Theano.

7.3. Model Refinement

We adopt grid search strategy to find the best possible hyperparameters for RNN and Recursive Neural Networks

7.4. Model Evaluation

We evaluate and compare the performance of the three models in terms of accuracy on classifying sentiment of movie reviews. We pay close attention on negative samples since this is where the RNN and Recursive Neural Networks may outperform the Naïve Bayes model.

8. Reference:

[1] Pang and L. Lee. 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1–135.
[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[3] Socher, R & Perelygin, A & Wu, J.Y. & Chuang, J & Manning, C.D. & Ng, A.Y. & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. EMNLP. 1631. 1631-1642.
[4] https://nlp.stanford.edu/sentiment/treebank.html
[5] https://www.mturk.com/
[6] http://scikit-learn.org/stable/
[7] https://www.tensorflow.org/
[8] http://deeplearning.net/software/theano/