Paper 5 | Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures by Aditya Timmaraju and Vikesh Khanna – published in 2015

Abstract

1. Using Recursive Neural Networks along with Recurrent Neural Network (where each time step is a sentence in the review) to classify paragraphs of text (multiple sentences in the case of movie reviews)
2. Through experiments, the paper shows that hand-crafted features carry complementary discriminative information in addition to that present in the learned neural network models

Related work

1. Traditionally, sentiment classification has been solved using linear classification methods such as SVM and logistic regression
2. In early work, Naïve Bayes classification and Maximum Entropy were studied
3. **Maas et. al propose a strategy to learn word vectors specifically for sentiment analysis and their semantic + sentiment model captures sentimental similarity between words extremely well**
4. Richard et. al argue that semantic word spaces can't express the meaning of longer phrases in a principled way. They introduced Recursive Neural Tensor Network that pushes that state of the art in single sentence binary classification and are able to capture phrase-level sentiment information. They also introduced 'SST' dataset

Model variations

1. Using semantic word vectors and recurrent architecture (Method 3.1)
    - In each review, split sentences from one another
    - Represent each sentence with the average of word vectors of each word in the sentence
    - Feed these averaged word vectors into a Recurrent Neural Network (RecNN), which is designed in such a way that the cross-entropy loss only propagates from the last time step, all the way back to the first-time step
2. Recursive Neural Network with mean likelihood (Method 3.2)
    - Split review into sentences
    - Feed the parse tree of each sentence into a recursive neural network
    - The class probabilities output by the RNN are averaged to decide the most likely class
3. Recursive Neural Network with affine NN (Method 3.3)
    - Same as model 2 but the hidden vectors output by RNNs are averaged into a single vector and fed into a 2-layer NN
4. Averaged semantic word vectors (Method 3.4)
    - Represent each review by the average of the word vectors of all the words in the review
    - Use the averaged word vector as a feature to classify the sentiment of the review using an affine NN/SVM

5. Averaged Semantic word vectors and BoW features (Method 3.5)
   - In addition to model 4, the paper also added BoW features with IDF weighting
   - Use an SVM classifier on this concatenated feature vector
   - This is the best performing model
6. **Recursive-recurrent neural network architecture** (Method 3.6)
   - Use the concept of recursively learning phrase-level sentiments for each sentence and apply that to longer documents (forming sentiment opinion from left to right)
   - Split movie review into sentences
   - Each sentence is fed into recursive neural network (RNN), which outputs a hidden vector and a sentiment for the sentence
   - Pass the RNN's hidden vector of each sentence as an input to a recurrent neural network (RCNN) – each sentence is considered a time step
   - It is conceivable that sentences that occur early on in a review need to be considered differently from those in the last part in order to evaluate their importance for sentiment classification
   - This model allows us to capture phrase-level sentiment for each sentence and sentence-level sentiments for the whole document

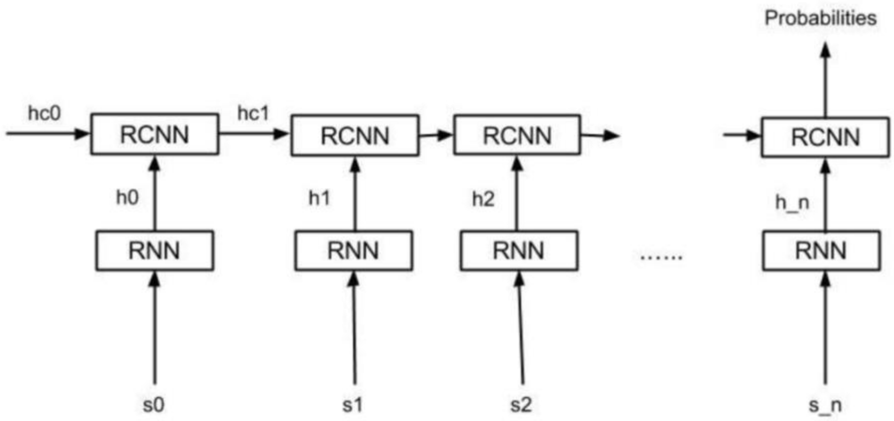

Figure 1: Approach 6: Classification Architecture.

7. Data
   - IMDB movie review dataset
   - Train the word vectors using the skip-gram architecture (use Word2Vec)
8. **Transfer learning with RNNs**
   - Trained a recursive neural network using the SST dataset. The dataset contains fine-grained phrase-level sentiment labels (5 classes)
   - The paper used this trained model on the classification task on the IMDB movie review. The authors do not update the RNN parameters during training as the IMDB dataset does not have fine-grained labels
   - Used NLTK tokeniser (punkt) to split movie reviews into sentences and Stanford parser to generate its parse tree

Results

Approach	Test Accuracy
Method 3.1, Mean Word - RecNN	82.35%
Method 3.2, Mean Prob - RNN	81.8%
Method 3.3, RNN-Affine	81.42%
Method 3.4, SVM-RBF	76.69%
Method 3.4, SVM-Linear	83.28%
Method 3.5, SVM-Linear	**86.496%**
Method 3.5, 2-layer NN	83.94 %
Method 3.6, RecNN-RNN	83.88%

Table 1: Summary of results

1. In the BoW model, the paper used IDF weighting, removed stop words, and performed L2 normalisation
2. Baseline model (method 3.4)
3. The NN uses sigmoid activation and cross-entropy loss function
4. For RecNN-RNN, the paper experimented with reversing the order of the sentences input but that did not have any significant impact on the test accuracy (83.76% vs 83.88%). Both variations have a learning rate of 0.001 but differ in regularisation strength (0.00001 and 0.00003)

Conclusion

1. Performed several experiments with approaches that have traditionally been used for sentiment analysis, like SVM/Affine neural networks
2. Extensively experimented with the proposed architecture – **Recursive Neural Network for sentence-level analysis and a recurrent neural network on top for passage analysis**
3. The paper observed that the information in the semantic word vectors is complementary to that combined in the TF-IDF BoW representation, since addition of these features to the averaged semantic vectors improved test accuracy by 3.2%
4. An important observation is that **the proposed model can perform almost as well as featured-engineered models** (86.50% vs 83.88%). This shows what the model can achieve and with more architectural and fine-tuning, the model can possibly outperform the state-of-the-art